KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Reliable and Low Delay Communication for Wireless Sensor Networks using In-Band Full Duplex

Tom Vermeulen

Supervisor:
Prof. dr. ir. Sofie Pollin

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

September 2017

Reliable and Low Delay Communication for Wireless Sensor Networks using In-Band Full Duplex

Tom VERMEULEN

Examination committee:
Prof. dr. ir. Hugo Hens, chair
Prof. dr. ir. Sofie Pollin, supervisor
Prof. dr. ir. Marian Verhelst
Prof. dr. ir. Liesbet Van der Perre
dr. ir. Alessandro Chiumento
Prof. dr. ir. Ingrid Moerman
(UGent)
ir. Andre Bourdoux
(imec)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

September 2017

Acknowledgments

I would like to start by thanking the first person who believed in me during my PhD, even before I was convinced. Sofie, the past few years have been an amazing experience working for you. You are a great leader and mentor and I've learned a lot from you. I sincerely would like to thank you for all the advice you've giving me and for allowing me to grow to become the person that I am today. I could not have wished for a better supervisor than you.

Besides Sofie, I would like to thank the whole networked systems team, Alessandro, Andrea, Bertold, Brecht, Cheng-Ming, Hazem, Jona, Mahdi, Ning, Qing, Seyed Ali, Sreeraj, Steven, Yuri and Zhenya for all the help and discussions but probably more importantly for all the great activities, lunch breaks and coffee breaks. I would especially like to thank Bertold for always giving me good advice when I was stuck on a problem. And of course the networked systems triathlon team, Brecht and Cheng-Ming, for the trainings and sport events.

At our electrical engineering department, I have the luxury of calling a lot of my colleagues also my friends. There are too many to name but I'd like to thank a few people in particular. Otto, thanks for always being there for me, not only during work but more importantly outside of work. Steven, thank you for being such a great friend and often dropping your current work just to help me. Nico, thanks for providing me with an often needed caffeine shot and for the nice chats. And finally, Fernando, thank you for the wonderful collaboration and the inspiring talks.

Next to my PhD activities, I was also involved in education related activities. Together with Otto and David and later Steven, I founded TechStart to push more entrepreneurial activities into the engineering curriculum. It was great working together with you guys, thanks for all our meetings and brainstorming activities. I would also like to thank LCIE for supporting us to make TechStart even more succesful. Next, I was also involved in Sagio.be through the board of directors. Over the years, I've learnt a lot working for this organisation and I

would therefore like to thank all the volunteers who make Sagio great.

From imec, I'd like to thank all the people from the PERSYSBE group to allow me to use their hardware and their knowledge. I want to especially thank Barend van Liempd for the many discussions and explanations. Without your help this work would not be of the same quality.

During my research stay at the University of California, Los Angeles (UCLA), I was lucky to meet a great group of people. First of all, Prof. Danijela Cabric, thank you for letting me work in your research group and thank you for all the great advice and meetings. Furthermore, I would like to thank Mihir, Ghaith, Shailesh, Jingwen and Han for showing me around campus and Los Angeles.

I'd also like to thank my doctoral committee. Thank you prof. Hugo Hens, prof. Marian Verhelst, Prof. Liesbet Van der Perre, Prof. Ingrid Moerman, dr. Alessandro Chiumento and Andre Bourdoux, for all the discussions and for improving this work. I would also like to thank the Agency for Innovation by Science and Technology in Flanders (IWT) for providing me the funding to complete this PhD.

I'm so grateful to have a wonderful group of friends. Jimmy, thank you for being my best man and for believing in me. Simon, thank you for being the greatest great-grand nephew there is and for always being there to help. Kurt, thank you for always making me laugh and for thinking the same way as I do. Glenn and Jonas, thank you for spending so much time with me on the bike. Finally, I'd also like to thank Bart, Bart, Cedric, Maarten, Pieter, Simon, Sven, Thomas and Wouter for always being there and ready to have a good time.

Laat me nu even naar het Nederlands overschakelen om nog een aantal speciale mensen te bedanken. Mama, papa, bon en bompa, de laatste jaren zijn misschien niet altijd even gemakkelijk geweest op vlak van gezondheid. Maar toch was er altijd 1 constante in jullie leven, jullie onvoorwaardelijke liefde voor mij. Een doctoraat was voor jullie niet altijd even duidelijk, "Neen, ik ga niet meer naar school" en "Neen, ik zit niet meer in een klas, ik sta nu zelfs soms voor de klas". Hoe onbegrijpelijk het ook voor jullie was, jullie bleven mij steunen. Jullie fierheid is wat mij vaak door de lastigere periodes hielp.

Daarnaast heb ik er sinds kort nog een fantastische tweede (schoon)familie bijgekregen. Marjan, Walter en Tom, bedankt om mij met warmte te onthalen in jullie familie. Mattias, Simon en Cami, bedankt om de broers en zus te zijn die ik nooit gehad heb.

Tot slot is er nog de vrouw van mijn leven, Sarah. Met grote trotsheid mag ik je sinds kort ook effectief mijn vrouw noemen. Ik ben nog altijd niet zeker wat ik gedaan heb om jou te verdienen. De laatste 11 jaar met jou waren echt

onvergetelijk en buitengewoon fantastisch. Een groot man zei me ooit dat je lijstjes moet hebben in je leven en daarom had ik je graag bedankt voor

- er altijd voor mij te zijn;
- mij altijd te steunen;
- mij naar een hoger niveau te tillen;
- veel van mijn papers na te lezen;
- mee met mij te reizen naar exotische bestemmingen;
- mij gelukkig te maken;
- zo fantastisch te zijn.

Abstract

Our continuous demand for more wireless connectivity, higher data rates, lower latencies and improved mobility, ultimately leads to wireless congestion problems. As we connect more devices wirelessly to the Internet, the wireless spectrum becomes crowded, slowing down our wireless connections. Efficiently managing these wireless resources is key in enabling new applications such as the Internet-of-Things, smart homes and the Industry 4.0.

This PhD thesis presents a complete analysis and prototype of a novel enabling technology for future dense wireless networks. In-band full duplex (IBFD) allows wireless devices to simultaneously transmit data and acquire information on active surrounding devices. This provides continuous monitoring of the wireless spectrum, and more crucially allows to detect collision and interference in real-time, enabling efficient spectrum usage.

Four major milestones have been achieved in this work. First, a full system exploration of in-band full duplex for continuous spectrum monitoring is presented and analyzed, including a complete energy model and simulator for this new technology. Second, a feasibility study is presented concerning techniques to acquire continuous spectrum information, and more specifically to determine whether an interferer is present or not. Third, a flexible prototyping platform is developed which allows to run the new technologies in real-time. Finally, everything is implemented in real-time on a FPGA-based software defined radio. The prototype has IBFD-capabilities through the use of an electrical balance duplexer, and is able to detect collisions and interference in real-time using a statistical hypothesis test.

In this work, it is shown that IBFD-enabled spectrum monitoring can increase the throughput in dense networks with at least a factor two, and at the same time decrease energy consumption. Moreover, this PhD shows that it is possible to detect neighboring wireless devices at close distance with 100% accuracy and within 20 meters with at least 80% accuracy. Finally, experiments using a network of six prototypes validate the system exploration and show that significant gains both in throughput and energy consumption can be achieved.

Beknopte samenvatting

Onze continue vraag naar meer draadloze connectiviteit, hogere doorvoersnelheden en lagere vertragingen, leiden uiteindelijk tot een draadloze opstopping. Wanneer we meer apparaten draadloos met het internet verbinden, wordt het spectrum meer bevolkt, wat onze draadloze verbindingen vertraagt. Het efficiënt beheren van het spectrum speelt een sleutelrol in het mogelijk maken van nieuwe applicaties zoals het *Internet-of-Things*, slimme huizen en de Industrie 4.0.

Deze doctoraatsthesis stelt een volledige analyse en prototype van een nieuwe technologie voor dense draadloze netwerken voor. *In-band full duplex* (IBFD) laat draadloze apparaten toe tegelijk gegevens te versturen en informatie over naburige actieve apparaten te verkrijgen. Dit stelt hen in staat om continu het draadloze spectrum te controleren en onmiddellijk botsingen en interferentie te detecteren, waardoor het draadloze spectrum efficiënter gebruikt wordt.

Vier grote mijlpalen zijn er in dit werk bereikt. Ten eerste wordt er een volledige systeemexploratie van IBFD voor continue spectrumcontrole gepresenteerd en geanalyseerd, inclusief een energiemodel en simulator voor deze nieuwe technologie. Ten tweede wordt er een haalbaarheidsstudie gepresenteerd voor technieken om continue spectruminformatie te verkrijgen, en vast te stellen of er interferentie is. Ten derde wordt er een flexibel prototype platform ontwikkeld dat toelaat nieuwe technologieën efficiënt te ontwikkelen. Tot slot wordt alles op een FPGA gebaseerde *software defined radio* geïmplementeerd. Het prototype heeft IBFD mogelijkheden door het gebruik van een *electrical balance duplexer* en kan botsingen en interferentie onmiddellijk detecteren.

In dit doctoraat wordt aangetoond dat spectrum monitoring op basis van IBFD de datasnelheid met meer dan een factor twee verhoogt in dense netwerken, terwijl het de energieconsumptie verlaagt. Verder is het mogelijk om actieve draadloze apparaten in de nabije omgeving te detecteren met 100% nauwkeurigheid en binnen een straal van 20 meter met een nauwkeurigheid van minstens 80%. Tot slot tonen de experimenten met een netwerk van zes prototypes aan dat significante verbetering in zowel snelheid als energieconsumptie mogelijk zijn.

List of Abbreviations

6LoWPAN IPv6 over low power wireless personal area networks. 55, 68, 144, 148

ACK acknowledgment. 16, 143, 148

AD Anderson-Darling. 110, 128

ADC analog-to-digital convertor. 44, 47, 49

AP access point. 84, 95

ASIC application-specific integrated circuit. 176

BB baseband. 47

BER bit error rate. 68, 146

BTMA busy-tone multiple access. 23

CCA clear channel assessment. 16, 22, 23, 29, 51, 94, 141, 152, 160, 168

CDF cumulative distribution function. 62, 63, 67, 70, 122, 123, 164, 167, 175

CDP collision detection period. 24

CFO carrier frequency offset. 141, 142

CLAWS Cross-Layer Adaptable Wireless System. 133, 134, 138, 140, 143–150, 152

CRC cyclic redundancy check. 140, 142

CSMA carrier sense multiple access. 22, 24, 25, 80–82, 84, 118, 137, 156, 166

CSMA/CA carrier sense multiple access with collision avoidance. 16, 19, 22, 23, 26, 28, 29, 33, 51, 52, 56, 57, 66, 83–85, 97, 118, 136, 157–159

CSMA/CD carrier sense multiple access with collision detection. 21, 24, 118

CSMA/CN carrier sense multiple access with collision notification. 25, 31, 86

CTS clear-to-send. 23, 24

DAC digital-to-analog convertor. 44

DBTMA dual busy-tone multiple access. 24

DCF distributed coordination function. 22–24

EBD electrical balance duplexer. 48, 60, 61, 69, 74, 89, 99, 125, 165, 166, 171

ED energy detector. 110, 121

FCS frame check sequence. 143

FD-CSMA/CA full duplex carrier sense multiple access with collision avoidance. 84, 92–95, 97–104

FD-CSMA/CD full duplex carrier sense multiple access with collision detection. 29, 33, 83–85, 92–95, 97, 98, 101–104

FIR finite impulse response. 99, 102

FPGA field programmable gate array. 8, 10, 56–58, 60, 63, 64, 68, 71, 73, 75, 134–136, 138–141, 144, 156, 158, 165–168, 170, 171, 174–176

GPIO general purpose input output. 60, 61, 166

HD-CSMA/CA half duplex carrier sense multiple access with collision avoidance. 83, 84, 92, 93, 95, 97, 98, 100, 102–104

IBFD in-band full duplex. 7, 9, 10, 46, 48, 51, 52, 57, 62, 64, 67, 70, 72–74, 81, 82, 86, 88, 94, 97, 100, 110, 111, 117, 118, 120, 124, 125, 159, 164

ICMP internet control message protocol. 69, 147, 148

IMD intermodulation distortion. 46

IoT Internet of Things. 1, 3, 157

IPv6 internet protocol version 6. 69, 144, 146

IQ in-phase and quadrature. 6, 56

ISM industrial, scientific and medical. 4, 15

KP Kuiper. 110, 128

KS Kolmogorov-Smirnov. 110, 122, 123, 128

LNA low noise amplifier. 44, 47, 48

MAC medium access control. 6, 8–10, 15, 16, 21, 22, 25–27, 29, 41, 53, 55–58, 64, 66, 68–75, 80, 81, 92, 94, 118, 132, 134–141, 143–145, 148–150, 152, 153, 156–159, 161, 164–167, 172, 174, 175, 177

MACA-BI medium access with collision avoidance by invitation. 26

MIMO multiple input multiple output. 48, 74

MSK minimum shift keying. 141

NAV network allocation vector. 23

NF noise figure. 46

ns-3 network simulator 3. 19, 93–95, 104, 108, 119, 159–161

OQPSK offset quadrature phase-shift keying. 140, 142

ORBIT Open-Access Research Testbed for Next-Generation Wireless Networks. 55, 137

OSI Open Systems Interconnection. 6

PA power amplifier. 48

PHY physical. 6, 8, 9, 15, 18, 41, 53, 55–58, 64, 68, 69, 71, 74, 75, 88, 94, 132–141, 143–146, 148–150, 152, 156, 158, 164, 166–168, 172, 174, 176, 177

PLL phase locked loop. 32, 149, 150

PPDU PLCP protocol data unit. 140

PSDU physical layer service data unit. 140

PSO particle swarm optimizer. 58, 60, 61, 166

QoS quality of service. 2, 25, 26

QPSK quadrature phase-shift keying. 44, 89, 124, 129

RACK real-time acknowledgment. 85, 86

RPL IPv6 routing protocol for low power and lossy networks. 146–148

RSSI received signal strength indicator. 141

RTS request-to-send. 23, 24

SDR software defined radio. 6, 8–10, 56–59, 68–70, 72, 132, 134–140, 143, 144, 149, 152, 153, 156

SI self-interference. 43, 44, 46–50, 52, 61, 81, 82, 88, 89, 110, 111, 113, 115, 121, 124, 128, 162, 164

SIC self-interference cancellation. 47, 99, 101, 111, 114, 115

SINR signal-to-interference-and-noise-ratio. 16

SNR signal-to-noise ratio. 18, 33, 48, 52, 68, 88, 89, 100, 146, 163

TCP transport control protocol. 138

TCP/IP transport control protocol/internet prococol. 68, 144

TSCH time-slotted channel hopping. 26

USRP universal software radio peripheral. 56–58, 60, 61, 63, 64, 68, 74, 88, 89, 110, 125, 135, 136, 139, 145, 158, 164, 166–168, 171, 175, 177

WARP Wireless open-access research platform. 57

List of Symbols

α	Full duplex power ratio
$\bar{\gamma}$	Signal-to-self-interference-and-noise ratio
$\bar{\mathcal{E}}_\mathrm{b}$	Average energy per bit
$\bar{\mathcal{E}}_\mathrm{b}^\mathrm{FD}$	Average energy per bit for full duplex transmissions
$\bar{\mathcal{E}}_\mathrm{b}^\mathrm{HD}$	Average energy per bit for half duplex transmissions
$\bar{\rho}_\mathrm{c}$	Average number of retransmissions due to collisions
$\bar{\rho}_\mathrm{i}$	Average number of retransmissions due to interference
$\bar{\tau}$	Average number of transmission trials
$\bar{\tau}_\mathrm{d}$	Average number of transmission trials due to decoding errors
$\bar{\mathcal{E}}_\mathrm{b}$	Average energy per bit
γ_c	Reduction in time per bit due to collision detection
γ_i	Reduction in time per bit due to interference detection
$\mathbb{P}_\mathrm{SD}$	System-level detection probability
$\mathbb{P}_\mathrm{SF}$	System-level false-alarm probability
$\mathcal{H}_0$	Null hypothesis
$\mathcal{H}_1$	Alternative hypothesis
$\mathcal{T}$	Throughput
b	Number of bits per symbol
L_H	Length of the header in bits

L_{O}	Length of the overhead in bits
L_{P}	Length of the payload in bits
P	Power consumption
P_{n}	Noise power
P_{PLL}	Power consumption of the phase locked loop circuit
P_{RXTX}	Power consumption of the full duplex transceiver state
P_{RX}	Power consumption of the receive state
P_{TX}	Power consumption of the transmit state
R_{s}	Symbol rate
T_{b}	Time per bit
T_{c}	Time per bit to detect a collision
T_{H}	Transmit time of the header
T_{i}	Time per bit to detect interference
T_{O}	Transmit time of the overhead
T_{P}	Transmit time of the payload

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Wireless traffic jam expected

In the beginning of 2017 Belgium had just over 11 million people. 91% of these people own a wirelessly-connected computer and 59% owns a smartphone [1]. Now imagine all these people connecting their dishwasher, TV, heating system, and so on, to the Internet. Not only at home, also in factories or warehouses, like in Amazon's robot warehouse [2] or on the road like the autonomous cars from Google [3], more and more 'things' are becoming connected, this is what we call the Internet of Things (IoT). Gartner [4] expects the number of connected things to grow by 220% by 2020 compared to 2016, as shown in Figure 1.1. Cisco [5] on the other hand estimates that by 2020 there will be 12 billion machine-to-machine connections, which is more than double the number of connections today. Moreover, Cisco estimates 600 million wearables and 3.1 billion Internet-enabled televisions worldwide. 50% of the Internet traffic generated by these devices will go through WiFi networks, with another 16% going through the cellular network, therefore, the majority of the traffic will be transmitted wirelessly. On top of this, the average speed per device will more than double by 2020 [5].

Most of this traffic, around 80% will come from streaming video services, especially with the move towards higher resolutions [5]. Smaller portions will come from browsing and sharing data. With all these applications combined, the average Internet user will generate up to 44 gigabytes per month. This equals a total of 2.3 zettabytes, i.e., 10^{21} bytes, of transmitted data bits globally during the whole year. Needless to say, current wireless networks and infrastructures

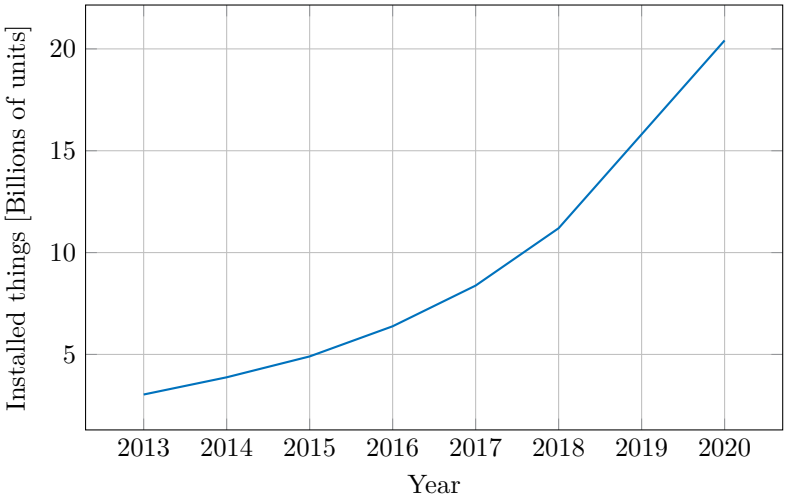


Figure 1.1: The installed number of 'things' will increase rapidly in the coming years. (Source: [4])

are not made to handle this huge amount of data generated by these massive amount of devices.

The quality of service (QoS) requirement of this massive amount of devices is also increasing, fueled by new applications, each with their own diverse requirements. An example of a group of such applications is called the Tactile Internet [6]. Each of these applications requires extremely low delay (<1 ms) and high reliability (>99.99%). Applications include fully autonomous cars who communicate with each other in order to drive close to each other and brake together. Another example is virtual reality which requires very minimal delay between the movement of the head or arms and the image on the screen. These applications are currently not yet possible, especially not when considering the constraints of wireless communication.

Another great set of applications, which is already being rolled out at the moment, relates to the notion of smart cities. Figure 1.2 gives an overview of the applications, detailing the different subcategories and their current installed base percentage. The total number of installed things within smart cities was around 1.6 billion in 2016. In smart cities a multitude of sensors is used to measure and gain insights into several processes. For example, sensors can be used to continuously monitor air quality or water pollution and allow regulators to take action when things go wrong. In big sites like industrial zones and office areas, the energy efficiency can be monitored. Gartner estimates that in these

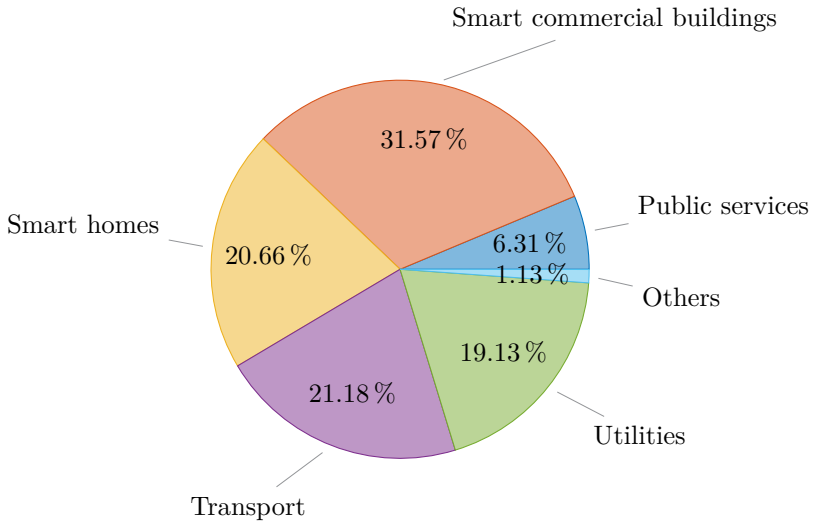


Figure 1.2: Overview of installed connected things within smart cities in 2016. (Source: [7])

big sites the IoT can save up to 30% in the cost of energy, spatial management and building maintenance. In smart homes, the biggest applications are smart TVs, smart thermostats and smart light bulbs. Gartner expects that this category will grow the most in the coming years. Each of these devices does not have high requirements, however, when combined these things require vast amounts of data throughput for their operation.

Finally, some applications require an extremely low energy consumption because they are battery powered. For example, the wireless sensor networks in the smart city example might not be easily accessible. In these remote locations it is cumbersome to replace the battery each month or even each year. These applications require battery lifetimes of around 10 years in order to make them cost effective and easy to maintain. Not only wireless sensor networks benefit from energy-efficient communication, in fact every battery powered device like your smartphone or your laptop can benefit from this, extending the battery lifetime from a few hours to a full day or even two. Besides in battery-powered devices, lowering the energy consumption is also essential from an environmental point of view. The International Energy Agency projects that in 2017 network-enabled devices will consume over 800 TWh globally [8]. That is more than ten times the overall domestic energy usage in Belgium [9].

One thing that should be clear from the above discussion, is that each of

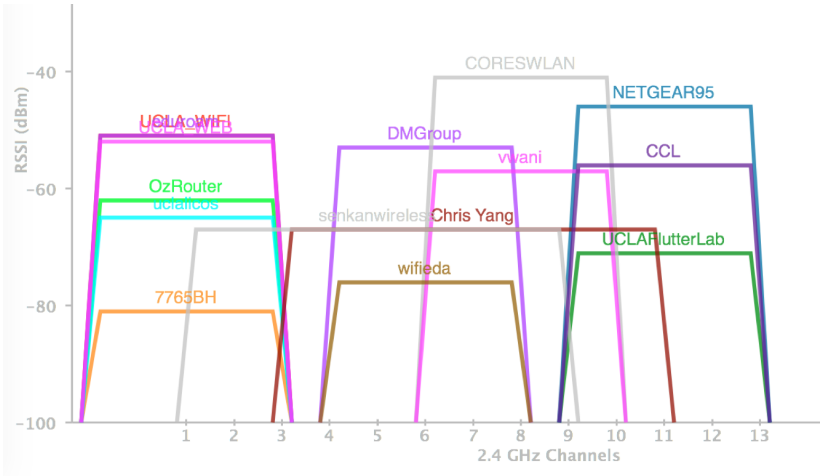


Figure 1.3: WiFi spectrum usage at an office location on the UCLA campus.

these applications have **different requirements**. Some need high throughput, others need extremely low delay and high reliability and others need none of the above but require a low energy consumption. On its own achieving these requirements will be challenging. The problem is even more challenging as all these applications will need to share the same networks. Each network will therefore need to be able to give some devices a highly reliable service while others need a high throughput service for example, all while taking the energy consumption into account.

This all comes at a point where the physical medium, i.e., the wireless spectrum, is getting more and more crowded. An important example is the 2.4 GHz industrial, scientific and medical (ISM) band which is used, among other things, for WiFi communication. Figure 1.3 shows a scan of the nearby WiFi access points at an office location at UCLA. The whole band is used, with each access point overlapping with at least five other access point. Not only the 2.4 GHz ISM band is heavily used, also other bands, especially below 1 GHz have a high utilization factor with more than 60% utilization being reported in [10].

Because of the limited amount of free wireless spectrum, it is difficult to meet the requirements for the diverse applications discussed above. The problem arises from too many devices with demanding applications trying to access the same wireless spectrum and essentially collapsing the network. Currently, most devices first sense the wireless medium for any other devices already transmitting in the vicinity. If no other device is detected, the device will transmit its data. However, in crowded places, there is a high probability that

two or more devices will decide to transmit data at the same time, causing a wireless collision. During the collision, the radio waves from each of the transmitters are added together at the receiver, essentially corrupting the data of all the current transmitters. Because of this collision the same data needs to be transmitted a second or even a third time before it is received correctly. This means that devices are more often in a power hungry transmit or receive state than in a low power idle state. Therefore, batteries die faster and we have to charge our devices or replace the batteries more often. Also the delay and throughput are affected as it takes longer to actually transmit and receive the data.

New promising techniques are being developed that potentially solve the problems at hand. One of these techniques is in-band full duplex [11] which enables wireless devices to simultaneously use their transmitter and receiver chain on the same frequency. The basic premise of in-band full duplex is that devices can double their bi-directional throughput by transmitting and receiving data at the same time over the same link. However, we believe that in-band full duplex can do more than that, leading to the following hypotheses of this thesis.

Hypotheses of this thesis
It is possible to use in-band full duplex to improve wireless congestion in dense heterogeneous networks, essentially improving delay, throughput and energy consumption altogether.

1.2 Scope of this thesis

This doctoral work provides a significant contribution towards spectrum and energy waste-free communication, enabling efficient use of the spectrum by many, possibly heterogeneous, devices. The main contributions of this doctoral work are techniques to solve the congestion problem in crowded areas exploiting in-band full duplex. We therefore focus on the underlying problem, which is that transmitters currently cannot detect other colliding transmitters and interference during a transmission, i.e., acquire context information. As such, valuable wireless spectrum is wasted when a transmission is corrupted. Shortening this spectrum waste is crucial in achieving the requirements for next-generation wireless communication: high spectral efficiency, high reliability, low latency and low energy consumption. More specifically, this thesis analyzes and implements techniques to increase throughput, decrease delay and decrease energy consumption all at the same time in very dense networks.

The congestion problem can be split into four research questions that will be answered throughout this work:

- **Research question 1:** Assume a device is able to acquire instantaneous context information, how can we then create low latency communication without wasting spectrum and energy?
- **Research question 2:** How can we enable wireless devices to simultaneously transmit and acquire context and contention information on the same frequency?
- **Research question 3:** Can we build a real-time prototype wireless node which uses our novel techniques, requiring changes in the physical and medium access control (MAC) layer?
- **Research question 4:** Can we validate the performance of these prototypes in a connected network with multiple radios?

The techniques in this work focus on the physical (PHY) and MAC layer of the Open Systems Interconnection (OSI) model. Moreover, cross-layer optimizations are used to coordinate both layers to solve the problem at hand. The PHY layer defines the physical specifications of the connection. In this thesis some PHY entities like the in-phase and quadrature (IQ) samples are used to optimize the MAC layer. The MAC layer defines how devices can access the wireless spectrum.

1.3 Methodology and key contributions

The remainder of this work will first describe the **system exploration**, and then a **validation** in a dense software defined radio (SDR) network, as visualized in Figure 1.4. First, the system exploration investigates how throughput, delay and energy cost can be improved in very dense networks, exploiting in-band full duplex. Next, enabling techniques are analyzed through a feasibility study, which bridges the gap between the theoretical analysis and the practical validation. Finally, in the validation phase, extensive network experiments are performed to confirm the results of the exploration phase.

Following this exploration-validation flow, the four research questions from the previous section are answered with four key contributions of this doctoral work.

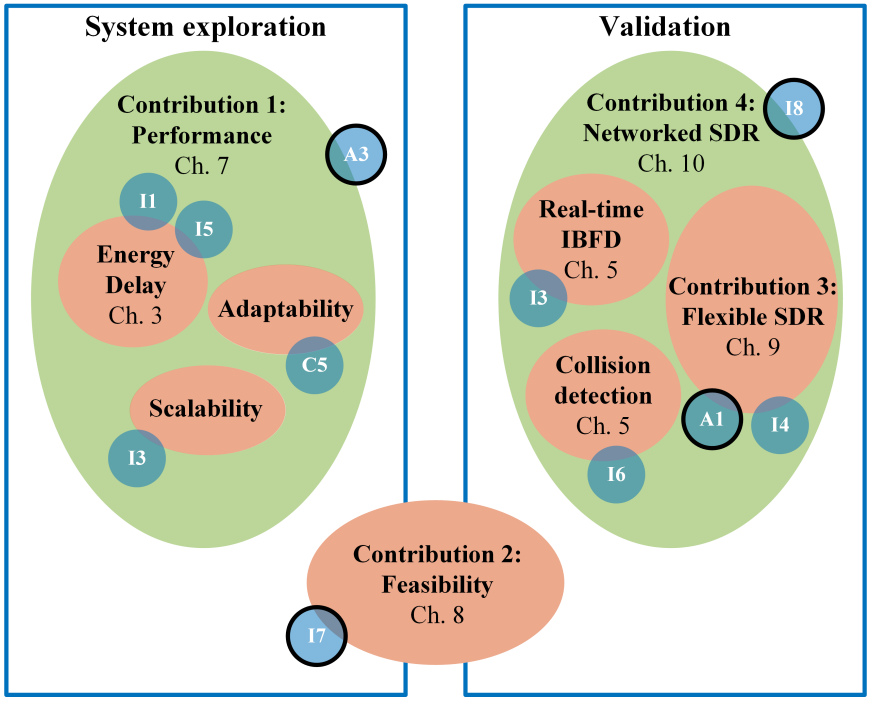


Figure 1.4: The flow followed in this thesis together with references to the key contributions and publications (cf. Section 1.5). Publications bundled in the second part of this doctoral work are surrounded by a black line.

Contribution 1: System exploration, exploiting in-band full duplex for performance improvements

Currently, devices either transmit or receive on the same frequency but not both at the same time. However, if a device would be able to acquire instantaneous information about what is happening in the environment, it would be possible to completely eliminate the spectrum waste. Therefore, a thorough performance analysis when using in-band full duplex (IBFD) for sensing and transmitting at the same time is performed. A transmitter can then detect neighboring devices and abort its transmission to avoid spectrum waste.

To benchmark the proposed protocol against the current state of the art, a novel energy model for in-band full duplex was designed. The model contains both the state of the art in half duplex communication as well as the state of the art in in-band full duplex communication. This allows to prove from a theoretical

point of view that sensing during a transmission, enabled by in-band full duplex, not only increases throughput but also decreases energy consumption. The model and the analysis are detailed in Chapter 7.

Contribution 2: Feasibility study, enabling techniques for instantaneous contention information

To prove the feasibility of the analyzed technology, we continue with a measurement-based exploration of techniques to enable sensing at the transmitter. More specifically, the proposed detection techniques allow transmitters to detect other transmitters during a transmission. These techniques are enabled by in-band full duplex which cancels the self-transmitted signal. For the digital processing, the sensing relies on statistical hypothesis testing.

The results show that it is possible to detect neighboring devices with a high probability within 20 meters of the transmitters. At 20 meters this theoretically doubles the throughput, while at short distances the throughput can be increased five-fold. The feasibility study is presented in Chapter 8.

Contribution 3: Prototyping, a flexible platform for real-time cross-layer experimentation

In order to prototype the sensing techniques from the feasibility study, a field programmable gate array (FPGA) implementation on a SDR of the IEEE 802.15.4 [12] PHY layer and MAC layer was built. The implementation offers the necessary flexibility on all layers to add the necessary cross-layer changes. Furthermore, because everything runs on the FPGA, real-time performance can be achieved enabling network experiments.

The design has been verified and benchmarked against state-of-the-art commercially available hardware. It was found that the SDR implementation outperforms state of the art off-the-shelf radios in terms of real-time performance. The transmitter and receiver performance, measured in bit error rate, was shown to be on par with the commercially available hardware. Therefore, this platform is the ideal starting ground for benchmarking the novel sensing techniques developed in this work. The full implementation is detailed in Chapter 9.

Contribution 4: Validation, confirming results through network experiments

Finally, the SDR platform was augmented with in-band full duplex capabilities and sensing algorithms. The prototypes are able to detect neighboring transmitters in real-time during a transmission, allowing them to take immediate action if something goes wrong. The prototype is able to detect transmitters within 300 μs and can achieve a similar performance as was investigated during the feasibility study.

All SDRs were then wirelessly connected with each other to perform network experiments. To the best of our knowledge, this is the first time that network experiments with in-band full duplex were performed. The experiments validated the system exploration and show the significant gains which can be achieved. The prototype and network experiments are detailed in Chapter 10.

1.4 Outline

This doctoral thesis is split into two parts. The first part of this work presents an overview of the research field and some complementary results, while the second part presents four included papers.

Part 1: Overview of the research field and complementary results

Chapter 2 presents an overview of the research field. Some background information on dense networks and contention based channel access will be provided in order to present an in-depth problem statement on the problem defined in 1.2. Furthermore, a study of the current state of the art in terms of MAC protocols is provided.

Chapter 3 presents the proposed collision detection protocol and its corresponding energy model to compare it with the current state of the art. Moreover, the performance benefits of detecting collisions is presented here.

Chapter 4 explains the IBFD problem and why self-interference cancellation is needed. On top of this, a state of the art study is provided going deeper into all the different techniques to cancel the self-interference below the noise floor.

Chapter 5 zooms in on the different prototyping platforms wireless researchers use to validate and measure their protocols and implementation. More specifically, an overview of the platforms used by the PHY and MAC

communities is presented as well as some remarks for cross-layer research. Furthermore, an overview of the prototyping platform used in this PhD is presented in this chapter.

Chapter 6 concludes this doctoral thesis with an overall summary of the presented results as well as a detailed summary of the main papers. Furthermore, a look into the possible next steps for this work is also given.

Part 2: Included papers

Chapter 7 presents the novel MAC protocol for IBFD collision detection developed during this PhD. Moreover, an energy model is presented which allows to compare the novel solution with the state of the art.

Chapter 8 presents techniques to detect collisions using in-band full duplex. First, a theoretical overview of the different techniques is detailed, followed by a measurement-based analysis of these techniques.

Chapter 9 details the IEEE 802.15.4 implementation of a low power sensor node implemented on the FPGA of a SDR. The implementation is benchmarked against off-the-shelf available sensor nodes.

Chapter 10 details the implementation of the techniques from Chapter 8 on a SDR. Using these prototypes, performance benchmarks are presented by running these prototypes in a network.

1.5 Publications

Articles in international reviewed journals

A1 Van den Bergh B., **Vermeulen T.***, Verhelst M., Pollin S., "CLAWS: Cross-Layer Adaptable Wireless System Enabling Full Cross-Layer Experimentation on Real-Time Software-Defined 802.15.4", EURASIP Journal on Wireless Communications and Networking, vol. 2014, no. 1, pp. 187, 2014.

A2 Rajendran S., Van den Bergh B., **Vermeulen T.**, Pollin S., "IEEE 5G Spectrum Sharing Challenge: A Practical Evaluation of Learning and Feedback", IEEE Communications Magazine, vol. 54, no. 11, 2016.

*Equal contributor

A3 **Vermeulen T.**, Rosas F., Reynders B., Verhelst M., Pollin S., "Performance Analysis of In-Band Full Duplex Collision Detection", IEEE transactions on networking, submitted, 2017.

International conference papers (First Author)

I1 **Vermeulen T.**, Pollin S., "Energy-Delay Analysis of Full Duplex Wireless Communication for Sensor Networks", IEEE Global Communications Conference, pp. 455-460, 2014.

I2 **Vermeulen T.**, van Liempd B., Hershberg B., Pollin S., "Real-Time RF Self-Interference Cancellation for In-Band Full Duplex", IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), pp. 275-276, 2015.

I3 **Vermeulen T.**, Rosas F., van Liempd B., Verhelst M., Pollin S., "An Energy-Scalable In-Band Full Duplex Architecture", IEEE 20th International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), pp. 22-26, 2015.

I4 **Vermeulen T.**, Van den Bergh B., Pollin S., "Demo: A Software Defined Radio Platform for Rapid Cross-Layer Prototyping", Proceedings of the 2015 Workshop on Software Radio Implementation Forum (SRIF), pp. 1-4, 2015.

I5 **Vermeulen T.** Rosas F., Verhelst M., Pollin S., "Performance Analysis of In-Band Full Duplex Collision and Interference Detection in Dense Networks", IEEE 13th Annual Consumer Communications & Networking Conference (CCNC), pp. 595-601, 2016.

I6 **Vermeulen T.**, Laghate M., Hattab G., van Liempd B., Cabric D., Pollin S., "Nearly Instantaneous Collision and Interference Detection Using In-Band Full Duplex", IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), 2017.

I7 **Vermeulen T.**, Laghate M., Hattab G., Cabric D., Pollin S., "Towards Instantaneous Collision and Interference Detection using In-Band Full Duplex", IEEE International Conference on Computer Communications (INFOCOM), 2017.

I8 **Vermeulen T.**, Pollin S., "Sense and Abort: a Real-Time In-Band Full Duplex Collision Detection Network", IEEE International Conference on Computer Communications (INFOCOM), submitted, 2018.

Co-authored papers

C1 Pacheco L., **Vermeulen T.**, Pollin S., Solis P., "Evaluation of TSCH/IEEE 802.15.4e in a Domestic Network Environment", Internet of Things Infrastructures: Second International Summit, IoT 360 2015, pp. 257-262, 2015.

C2 Van den Bergh B., **Vermeulen T.**, Pollin S., "Analysis of Harmful Interference to and from Aerial IEEE 802.11 Systems", Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use (DroNet)", pp. 15-19, 2015.

C3 Li P., **Vermeulen T.**, Li H., Pollin S., "An Adaptive Channel Selection Scheme for Reliable TSCH-based Communication", International Symposium on Wireless Communication Systems (ISWCS), pp. 511-515, 2015.

C4 Pacheco L., **Vermeulen T.**, Pollin S., Solis P., "Performance Evaluation of TSCH in IEEE802.15.4e in the Context of the Internet of Things", XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2015.

C5 Reynders B., **Vermeulen T.**, Rosas F., Pollin S., "Adaptive In-band Full-Duplex Collision Detection for Balancing Sensing and Collision Costs", European Conference on Networks and Communcation (EuCNC), 2017

Part I

Overview of the research field and complementary results

Chapter 2

Congested wireless networks

In this chapter an overview of the research field is presented. First the overall congestion problem will be defined, both on the PHY and MAC layer, followed by the implications these have on the performance of the network. Next, a study of the state of the art is presented concerning improved MAC protocols.

2.1 Sources of performance loss in wireless communication

As discussed in the previous chapter, the number of wireless devices and the traffic keeps increasing and therefore the spectrum is being used more intensively. This creates unprecedented congestion problems in the wireless domain, due to collisions and interference. Ultimately, these contention problems result in higher latencies, higher energy consumption and lower throughput. A good example of heavily used frequency bands are the ISM bands around 2.45 GHz or 5.8 GHz. On these frequencies, many wireless technologies such as IEEE 802.11 [13] (WiFi), Bluetooth [14] and IEEE 802.15.4, operate simultaneously.

At the PHY layer, the main challenge is that these technologies simultaneously operate in the same frequency band, causing interference to each other. Interference is caused by the physical phenomenon of superposition of waves. When two wireless signals are transmitted simultaneously, the receiver will receive a superposition of the two signals. This phenomenon is illustrated in Figure 2.1 where two sine waves of different frequencies are added together. The resulting wave does not resemble any of the two original ones.

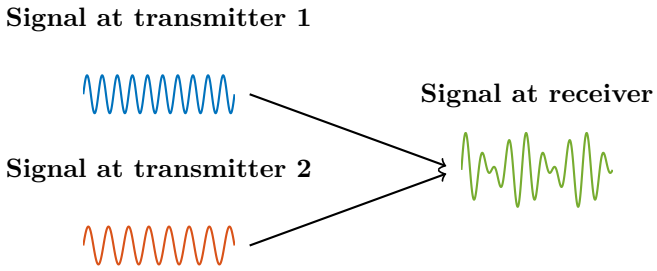


Figure 2.1: If two devices decide to transmit at the same moment, the receiver will receive a superposition of the two signals, causing a collision and corrupting the packet.

In order for data to be received correctly with these technologies, the signal-to-interference-and-noise-ratio (SINR), which is the ratio between the power of the first wave and the second wave, should be at least positive but preferably as high as possible. However, due to the superposition, the SINR could become small or even negative, effectively corrupting the data carried by the wave.

In theory, the MAC layer should avoid these superpositions by controlling the access to the medium. In practice however, this is difficult due to the way most devices access the medium. Typically, some form of carrier sense multiple access with collision avoidance (CSMA/CA) is used where each device that wants to transmit a packet will need to follow the steps defined in Figure 2.2. The node will first wait for a random time. This random backoff assures that it is unlikely that multiple nodes try to send simultaneously resulting in a collision. Next, the nodes performs a clear channel assessment (CCA), and if the medium is free, they will send their packet. If the channel is not free, the nodes will increase their backoff exponent to backoff even further. If the node has reached its maximum number of backoffs, it will report to the upper layers that the transmission has failed. If the packet transmission is finished, the node will wait for an acknowledgment (ACK), and if the ACK is received, the packet is successful.

This protocol has two main problems. First of all, the CCA mechanism is often tuned for a certain technology, which causes interference as different technologies can't hear each other [15] and will collide. Second, not only different technologies but also within the same technology, collisions are a problem, as the collision avoidance mechanism breaks down for a high number of devices [16].

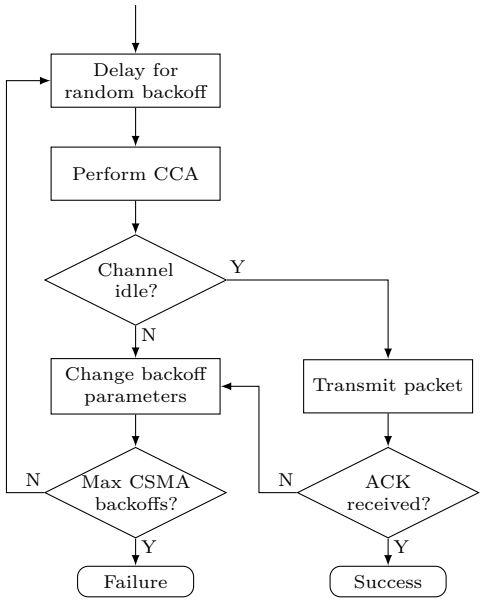


Figure 2.2: CSMA/CA algorithm found in most wireless technologies.

To illustrate these problems, let us formulate the average number of transmission trials ($\bar{\tau}$) required to correctly transmit a packet as,

$$\bar{\tau} = \bar{\tau}_{\mathrm{d}} + \bar{\rho}_{\mathrm{i}} + \bar{\rho}_{\mathrm{c}} \ . \tag{2.1}$$

Here $\bar{\rho}_{\mathrm{c}} \geq 0$ is the average number of retransmissions due to collisions with transmissions from the same network, and is therefore dependent on the number of active devices in the network. Typically, $\bar{\rho}_{\mathrm{c}}$ follows an exponential curve as function of the number of nodes [17]. $\bar{\rho}_{\mathrm{i}} \geq 0$ is the average number of retransmissions due to interference with transmissions from other networks under the condition of no collisions, and $\bar{\tau}_{\mathrm{d}} \geq 1$ is the average number of *transmission trials* needed to correctly decode a packet conditioned on the event of a reception without collisions or interference, i.e., retransmissions needed because of fading at physical layer.

In [17], the dependencies between $\bar{\tau}_{\mathrm{d}}$, $\bar{\rho}_{\mathrm{i}}$ and $\bar{\rho}_{\mathrm{c}}$ are derived. With $\bar{\tau}_{\mathrm{d}} = (1-\bar{P}_{\mathrm{f}})^{-1}$, one can rewrite $\bar{\rho}_{\mathrm{i}}$ and $\bar{\rho}_{\mathrm{c}}$ as follows

$$\bar{\rho}_{\mathrm{i}} = \frac{q_{\mathrm{i}}}{(1-q_{\mathrm{i}})(1-\bar{P}_{\mathrm{f}})}, \tag{2.2}$$

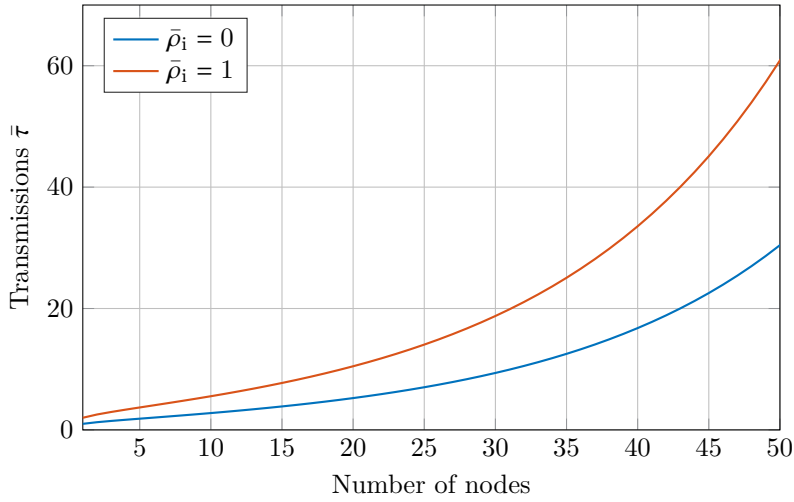


Figure 2.3: The average number of transmissions increases exponentially with the number of nodes.

$$\bar{\rho}_c = \frac{q_c}{(1-q_i)(1-q_c)(1-\bar{P}_f)}, \tag{2.3}$$

with q_i and q_c the interference and collision probability respectively. It is clear that an increase in interference or decoding errors also increases the number of collisions, as the medium becomes more congested.

Let us numerically evaluate (2.1) by using parameters from the IEEE 802.15.4 standard. For a 25 dB signal-to-noise ratio (SNR) link, this gives $\bar{\tau}_d = 2.2575$ [17]. The value is only dependent on PHY layer parameters and is therefore independent of the number nodes in the network. The average number of retransmissions due to collisions can be calculated by numerically solving the model from [18] with the parameters from the IEEE 802.15.4 standard, as detailed in [17]. Next, given this simple but accurate model, it is possible to plot the average number of transmissions trials ($\bar{\tau}$) as function of the number of nodes in certain scenarios, shown in Figure 2.3. The effect of collisions is clear, more active nodes means more congestion and therefore a higher probability of two or more nodes transmitting at the same time. $\bar{\rho}_i = 1$ means that 50% of the packets are corrupted due to interference as on average each packet is affected by one interference event, but some packets are affected with more. The distribution is an exponential distribution with mean 1, resulting in a 50% chance of no event during a packet. The effect of interference is less clear at first sight because it is independent of the number of nodes. However, an increase in $\bar{\rho}_i$, also increases $\bar{\rho}_c$ as more nodes will need to retransmit. One

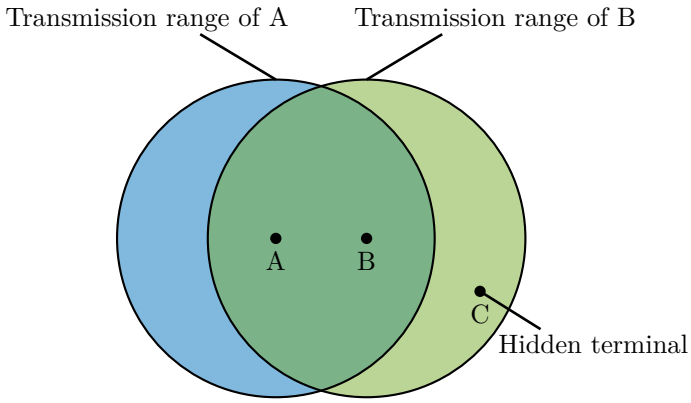


Figure 2.4: Illustration of the hidden terminal problem. Node C cannot detect any of the transmissions of node A.

of the key contributions of this thesis, discussed in Chapter 7, will elaborate on this. Looking more closely, we can see that $\bar{\rho}_{\mathrm{i}} = 1$ doubles the amount of retransmission irrespective of the number of nodes.

Not all collisions are caused by a failure of the CSMA/CA protocol, some are caused by hidden terminals. The problem is illustrated in Figure 2.4. Let us assume that node A wants to transmit to node B. Node A will therefore perform a clear channel assessment, determine that the spectrum is free and start transmitting to node B. During the transmission of node A, node C decides to also perform a clear channel assessment. As node C is not in the transmission range of node A, it will decide that the channel is free and transmit to node B. Both transmissions will arrive at node B and cause a collision. In [19], the effect of hidden terminals on the number of transmission trials is investigated. Figure 2.5 shows the increase in number of transmissions due to hidden terminals. If there are hidden terminals present in the network, the average number of transmissions increases much faster as the carrier sense mechanism fails.

Every retransmission increases the delay and energy consumption and decreases the throughput. By simulating a network of IEEE 802.15.4 nodes in network simulator 3 (ns-3) [20], we can show the effect of collisions on the throughput. Figure 2.6 shows the saturated throughput as function of the number of nodes. For a small number of nodes, the throughput goes up, this is a typical effect due to the random backoff. As more nodes are added to the network, the spectrum idle time decreases. However, due to an increased collision probability, the actual delivered throughput decreases significantly with more nodes. For

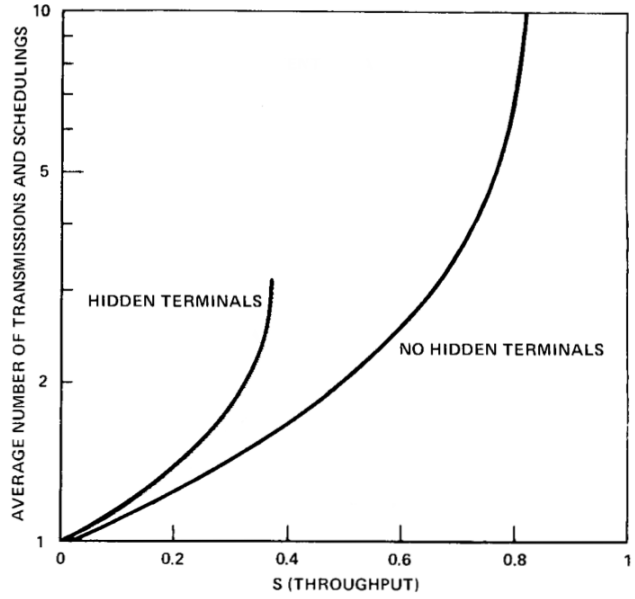


Figure 2.5: Hidden terminals cause more retransmissions as the carrier sense mechanism in the CSMA protocols fails. (Source: [19])

this particular simulation, the network loses 50% of its achievable throughput already at 60 nodes in the network, and 90% at 140 nodes in the network. Looking at the extreme case of 200 nodes, the throughput is almost 0.

For a lot of battery-powered applications, throughput is only a secondary requirement, the main objective there is energy efficiency. In order to quantify the energy performance, let us define the average energy per bit of a transmitter as

$$\bar{\mathcal{E}}_{\mathrm{b}} = PT_{\mathrm{b}}\bar{\tau}, \tag{2.4}$$

with P the power consumption of the transmitter, T_{b} the time per bit and $\bar{\tau}$ the average number of transmissions as defined in Equation (2.1). Using the energy model from [21] for an IEEE 802.15.4 device we can define $P = 30.67$ mW. T_{b} on the other hand is dependent on the physical layer settings, using the default settings for a packet length of 120 bytes, we can set $T_{\mathrm{b}} = 4.5778$ μs/bit.

Equation (2.4) allows us plot the average energy per bit in function of the number of nodes, shown in Figure 2.7. Assuming the amount of transmitted data remains the same, we see that compared to a network with only 1 node, i.e. no collisions, the battery life halves for a network of 30 nodes. Increasing

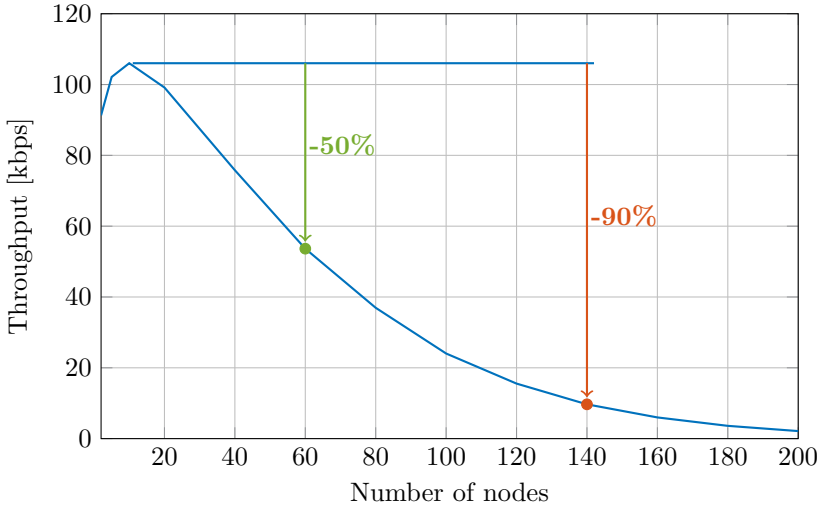


Figure 2.6: The throughput drops sharply with an increasing number of nodes, dropping 50% for 60 nodes and 90% for 140 nodes.

the network size even further, around 80 nodes the battery life per node is only 1/10th of the battery life without collisions. Therefore, mitigating the collision and interference problem is not only crucial for throughput but also for battery life. In the next section, we elaborate on the state-of-the-art approaches towards collision and interference mitigation.

2.2 State-of-the-art solutions

Over the years, several research groups have come up with partial solutions for these problems. In this section, we will focus on MAC protocols for devices in a star topology, where each device is connected to one central device. Ideally, one would like to detect collisions and abort the ongoing transmission instead of trying to avoid them before they happen, as this is shown to break down in dense networks. In wired networks for example, carrier sense multiple access with collision detection (CSMA/CD) is used to optimize the performance by detecting collisions during the transmission. This is possible because each device uses one cable to transmit and one to receive, essentially splitting the medium in two. In wireless systems, only one medium is present, and therefore other solutions are necessary. These solutions can be categorized in three sections: first, there are the general improvements of the collision avoidance algorithm in

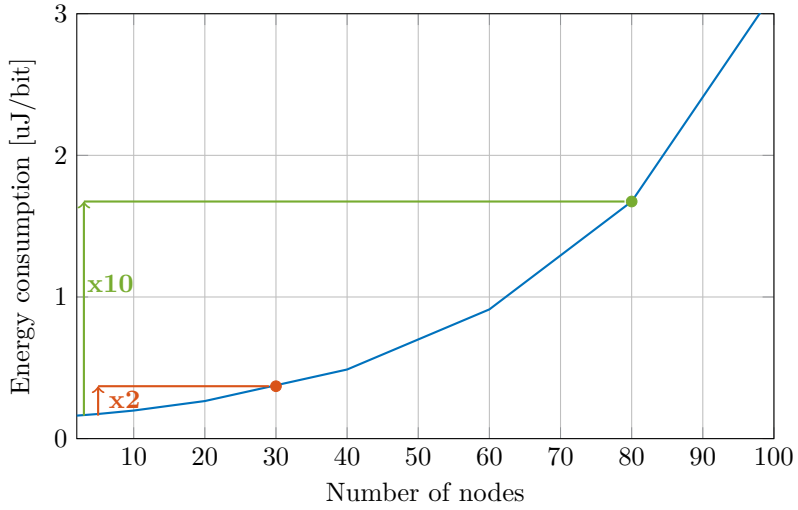


Figure 2.7: The energy consumption increases exponentially with the number of active nodes, doubling already at 30 nodes and increasing ten-fold at 80 nodes.

the CSMA/CA protocol, explained in Section 2.2.1; second, the solutions which add some form of collision detection to carrier sense multiple access (CSMA) in Section 2.2.2; and third, there are the non-CSMA MAC protocols, detailed in Section 2.2.3.

2.2.1 Collision avoidance and CSMA

The collision avoidance mechanism tries to avoid that two devices perform a clear channel assessment and therefore decide that the channel is free. To avoid this from happening, each device first waits a random time before performing the CCA. The interval of this random time defines the robustness of the mechanism. The bigger the interval, the lower the probability of two devices transmitting on the same time. However, increasing the interval has negative effects on the throughput and delay, as devices have to spend more time waiting before they can transmit data. Therefore, in this section an overview of the improvements to the collision avoidance mechanism are presented which do not increase the random backoff interval.

The first improvement is the IEEE 802.11 distributed coordination function (DCF) function [13] which uses in-band control frames to notify all surrounding nodes of an upcoming transmission. The protocol is explained in Figure 2.8.

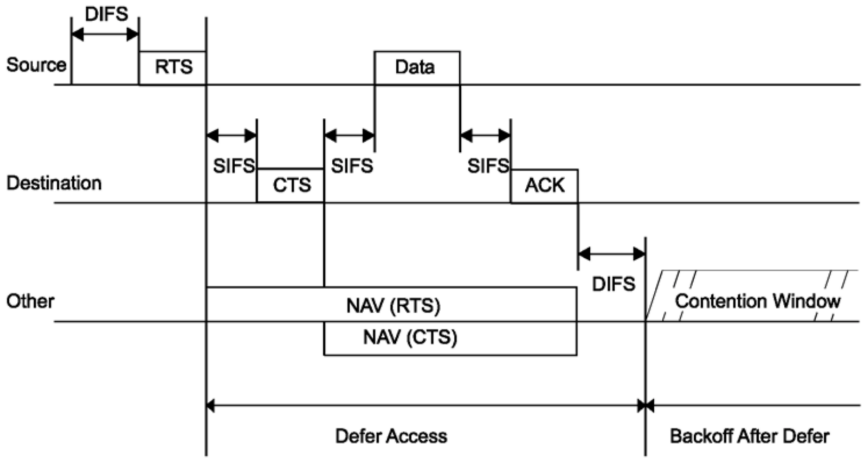


Figure 2.8: The RTS and CTS packets in the 802.11 DCF function allow all surrounding nodes to be notified of an upcoming transmission. (Source: [13])

First, the transmitting device will transmit a request-to-send (RTS) packet to the destination. If this packet is correctly received by the receiver, he will transmit a clear-to-send (CTS) packet. Based on the RTS packet, devices surrounding the source will update their network allocation vector (NAV) and defer access to the medium to avoid collisions. Based on the CTS packets, devices surrounding the destination will update their NAV. The DCF function avoids collisions and as shown in [18], avoids the decrease in overall throughput with an increasing number of nodes. However, in practice it is shown that the overhead due to this scheme significantly reduces the throughput [22]. In real-world applications, the throughput almost halves when enabling this feature, this leads most manufacturers to disable this feature by default. Moreover, interfering devices ignoring the DCF function still cause interference as they will not defer their transmission during the NAV.

Another solution is busy-tone multiple access (BTMA) [19], where the available spectrum is split into a data channel and a control channel. Before transmitting, the transmitter will listen on the control channel to determine if the medium is free, effectively replacing the CCA in the normal CSMA/CA protocol. If the channel is free, a busy-tone will be transmitted on the control channel and the data on the data channel. The main benefit of this solution is that the busy-tone can be designed in such a way that detection is fast and reliable. However, most of the problems from the previous section remain. In [23], an extension to BTMA is proposed which uses busy-tones in the control channel

simular to the RTS and CTS packets in the DCF function. In dual busy-tone multiple access (DBTMA), two distinct tones are used, one for the transmitter and one for the receiver. This way both nodes surrounding the transmitter and receiver are aware of the ongoing transmission. It is shown that DBTMA can double the network throughput when compared to the DCF function [23]. However, although that there is no overhead in the time domain, there is still some overhead in the frequency domain due to the control channel. Moreover, DBTMA only works if all devices are following the protocol.

The previous algorithms still transmit collisions in full if they occur, wasting time and energy. In [24], on out-of-band collision detection protocol is proposed based on DBTMA. As soon as the transmitter has determined the channel to be idle, it will transmit pulses in the control channel and data in the data channel. At the receiver side, the receiver will also transmit pulses in the control channel. The transmitter can therefore determine if a collision occurred early on. If the transmitter does not receive any pulses from the receiver or it receives pulses from another transmitter, it will abort its transmission and retry. Using this scheme, the throughput can be more than doubled. Unfortunately, only collisions can be detected and thus devices are still prone to interference from other technologies. On top of this, extra spectrum is needed for the control channel, which is already very scarce even for the data channel.

2.2.2 Collision detection and CSMA

In general, it is assumed that implementing the CSMA/CD protocol which is used in wired networks is not possible in wireless, as a transmitter cannot differentiate between its own transmission and a collision. However, it is possible to implement some form of collision detection in wireless which tries to emulate the performance of the wired CSMA/CD protocol. In this section, extensions to the CSMA protocol are detailed where collisions are detected in-band.

In [25], a wireless CSMA/CD (WCSMA/CD) scheme is proposed, where each transmitter will randomly stop its transmission for a small period of time, as indicated by the CD slot in Figure 2.9. During this slot, User1 for example senses for other ongoing transmissions. If a transmission is detected, User1 does not immediately abort its transmission but continues to transmit for a given collision detection period (CDP). This allows User2 and User3 to detect the collision as well. After the CDP, all nodes abort their transmission and try to acquire the medium again, as illustrated by the successful transmission of User2. It is clear that collisions can be shortened to the length of the CDP. Ideally this length should be as small as possible. However, this increases the probability of all nodes choosing the same random CD slot and therefore not detecting any

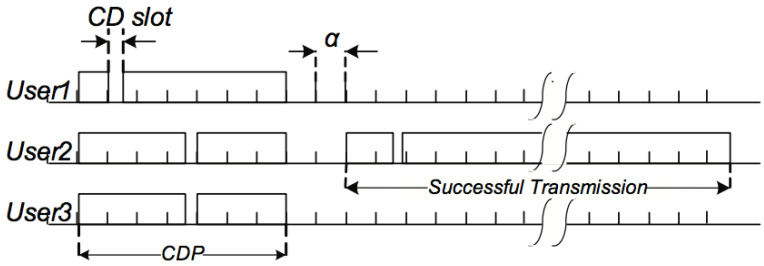


Figure 2.9: In the WCSMA/CD scheme each transmitter shortly stops transmitting to detect collisions. (Source: [25])

collisions. Moreover, in a practical system switching times between transmit and receive and back should not be ignored.

A second collision detection scheme called carrier sense multiple access with collision notification (CSMA/CN) is presented in [26]. In this scheme, the receiver tries to decide whether it is receiving a collision or a regular packet. The decoder at the receiver will decide whether the symbols it is getting are close to the ideal symbols or not. If the symbols deviate from the ideal symbols, it is classified as a collision and a notification is transmitted. Each receiver has a distinct signature allowing the transmitter to identify collisions at its receiver. The transmitter will continuously, during its transmission, correlate for this distinct signature and upon detection, will abort its transmission. They prove that they are able to detect 92% of all collisions. However, multiple problems arise which are not addressed in the paper, such as the fact that a collision can also interfere with the notification. On top of this, it is not clear how long it takes before the transmission is aborted, as the collision first needs to be detected by the receiver and then the transmitter needs to correlate and detect the notification.

2.2.3 Non-CSMA MAC protocols

The main benefit of the CSMA-based algorithms from the previous sections is flexibility, devices can come and go, and devices with lots of data can coexist in the same network with devices only transmitting a small amount of data. The drawback of CSMA is QoS, despite methods for collision avoidance and detection. Another class of MAC protocols, discussed in this section, focus on QoS, but are less flexible as they need a central control.

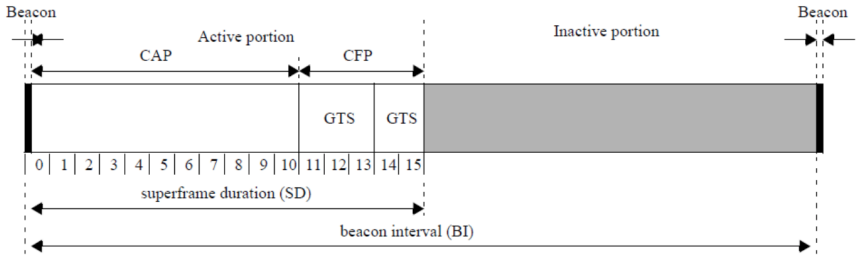


Figure 2.10: The superframe structure of the IEEE 802.15.4 beacon-enabled mode. (Source: [12])

First, let us start with a hybrid of both a centrally controlled protocol and a contention based protocol. The IEEE 802.15.4 standard defines a beacon-enabled access scheme as defined in Figure 2.10. The coordinator will periodically transmit a beacon to all connected devices for synchronization. The period between two beacons can be split in an active period and an inactive period based on the duty cycle. The active part is again split in two parts, in the contention access period (CAP), all nodes follow the CSMA/CA protocol as defined above. During this period, a device can send data to the coordinator but can also request a guaranteed time slot (GTS) to be used in the contention-free period (CFP). The CFP allows the coordinator to guarantee a certain QoS per device. For dense networks many of the problems discussed above still remains however.

In 2012 the IEEE 802.15.4e amendment [27] was introduced, focusing only on MAC layer improvements. One of these improvements is time-slotted channel hopping (TSCH), where time and frequency are split into slots, as indicated by the example in Figure 2.10. Each device is assigned certain timeslots and channels. Each of these slots is long enough to send the data and receive an acknowledgment. By using dedicated timeslots, collisions are avoided as no other devices are allowed to send then. By hopping from channel to channel, packet errors due to fading and interference can be avoided, as each device will use multiple frequencies and packets can therefore be retransmitted on a different frequency.

All previous MAC protocols were transmitter-initiated, meaning that the transmitter decides when to transmit, or were centrally controlled. It is also possible to let the receiver initiate the transmission. To do this, the receiver sends polling messages to all transmitters to request data. If the transmitter has data to send, it can respond immediately after the polling message and send its data. The medium access with collision avoidance by invitation (MACA-BI)

↓ Channel	Timeslots → 1	2	3	4	5	6
1	B → A					D → A
2			D → A		C → A	
3		C → A				
4				B → A		

Figure 2.11: Example of a TSCH schedule where each device is assigned a certain time and frequency slot.

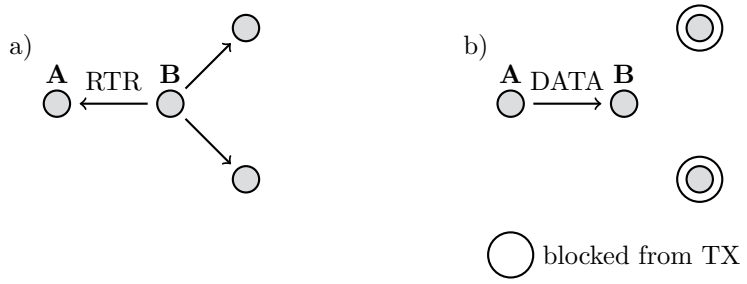


Figure 2.12: Using medium access with collision avoidance by invitation, the receiver polls the transmitters to request for data.

protocol [28], shown in Figure 2.12, uses such a scheme. In a first stage, the receiver **B** sends a request to receive (RTR) to **A**, all nodes surrounding **B** are blocked from transmission. In the second stage, **A** transmits its data to **B**. The main benefit of such a scheme is that the control overhead to avoid collisions is very short. However, this protocol only works in specific scenarios with predictable traffic pattern, as the receiver has to predict the traffic requirements. Therefore, in most situations, especially in bursty traffic, performance is worse than with transmitted-initiated schemes [29].

2.3 Conclusion

For contention-based systems the main sources of performance degradation are collisions and interference. Collisions come from inefficiencies in the MAC protocol, while interference comes from different technologies being unable to listen to each other. The result is that packets need to be retransmitted in order for the receiver to correctly decode them. A direct consequence is that the effective throughput goes down and the delay goes up. This means that

devices are more often in a power-hungry transmit or receive state and deplete their batteries faster or consume more power in general.

Some solutions were already proposed to reduce the number of collisions. Some try to optimize the current CSMA/CA algorithm, while others try to implement some sort of collision detection. However, all have one or more downsides. Either they use more resources in the time or frequency domain or they can only mitigate collisions and fail to solve the interference problem. To really solve the problem a true collision and interference detection protocol should be used in order to save energy and increase throughput, this will be investigated in the next chapter.

Chapter 3

Collision detection protocol and performance model

In this chapter, we will investigate how a collision detection protocol will look like and what its performance benefits are. For now, let us assume it is possible to simultaneously transmit data and detect collisions. Section 3.1 explains our novel collision detection MAC protocol. In Section 3.2, a novel energy model for such a transceiver is presented, followed by an energy and delay analysis in Section 3.3. Finally, in Section 3.4 an algorithm is analyzed which at runtime decides to turn on collision detection or not.

3.1 A collision detection MAC protocol

In this chapter, the full duplex carrier sense multiple access with collision detection (FD-CSMA/CD) MAC protocol is proposed, shown in Figure 3.1. The algorithm extends the CSMA/CA protocol explained in Chapter 2 with collision detection capabilities. First the node will try to acquire the medium by randomly backing off and performing a CCA. After that the node decides that the channel is idle and the transmission is started, the node will turn on its receiver and listen for an instantaneous acknowledgment coming from the receiver. If no acknowledgment is found after a certain time, the transmission is aborted and the node again backs off for a random delay. If, on the other hand, the instantaneous acknowledgment is found, the node keeps on transmitting until the transmission is finished. Figure 3.2 shows the packet transmissions in time. On the receiver-side, the node will, immediately after receiving the start

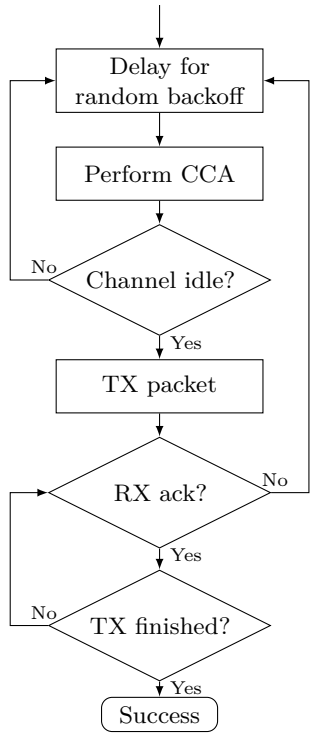


Figure 3.1: Flowchart of the FD-CSMA/CD algorithm (Source [17])

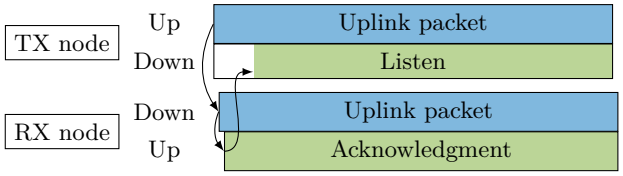


Figure 3.2: Full duplex collision detection allows to simultaneously transmit data and sense for collisions. (Source: [30])

of the packet, transmit an acknowledgment. However, if there is a collision or some interference the receiver will stop transmitting the acknowledgment to notify the receiver.

The protocol has multiple benefits. First, no extra resources are needed as the collision detection is done in-band and during the transmission. Moreover, if a collision occurs, the acknowledgment will be aborted. Therefore, the collision

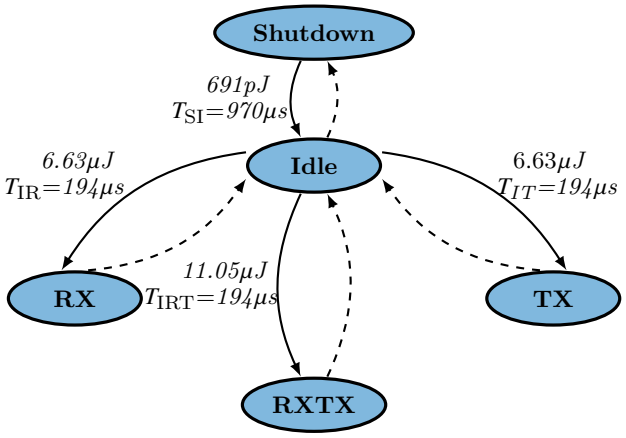


Figure 3.3: Transition diagram with transition energies and times (Source: [30])

does not interfere with the notification as was the case in CSMA/CN explained in Section 2.2.2. Finally, the acknowledgment can also be used to carry information, meaning that if the receiver has data to send to the transmitter, this can be done instead of the instantaneous acknowledgment without losing its purpose. Of course all these features come at a cost, each node will have to turn on both its transmitter and its receiver chain to enable the collision detection. However, in this work it is shown that this extra energy cost can be compensated for by all its benefits. The trade-off will be investigated in the following sections.

3.2 An energy model for collision detection

Before the energy trade-off can be investigated, we need to define a novel energy model for the proposed algorithm as there are currently no off-the-shelf radios that support this feature. The novel energy model is based on an IEEE 802.15.4 chipset, the TI CC2420 [31], which was measured and modeled in [21]. The model from [21] has four states: Shutdown (clock is turned off), Idle (clock is turned on), transmit (TX) and receive (RX). This is extended with a fifth state: full duplex (RXTX) (both receiver and transmitter are active). A node is in the full duplex state whenever it uses collision detection or transmits and receives a packet at the same time. The full state diagram is shown in Figure 3.3.

The power consumed in the first four states was measured in [21]. For the fifth state, we need to look into what is turned on in the transceiver. In this state, both the transmitter chain and the receiver chain of the transceiver will

Shutdown	Idle	RX	TX	RXTX
$144nW$	$712\mu W$	$35.28mW$	$30.67mW$	$56.95mW$

Table 3.1: Power consumption of the different states

be turned on, however, because both chains are operating at the same center frequency only one phase locked loop (PLL) can be used. To identify the power consumption of the PLL, we looked into a similar chipset [32] and identified P_{PLL} to be around 9 mW. Therefore, the power consumption of the full duplex state can be defined as

$$P_{\mathrm{RXTX}} = P_{\mathrm{RX}} + P_{\mathrm{TX}} - P_{\mathrm{PLL}}. \tag{3.1}$$

Table 3.1 gives an overview of the power consumption of the different states.

Switching between states also consumes energy, this is shown as well in Figure 3.3. The transition times and energies between the first four states is given in [21]. For the fifth state we use $E = TIV_{\mathrm{DD}}$ to calculate the energy, while the transition will be similar to the RX and TX state as most time is lost in locking the PLL. For completeness the transition times and energies are included in Figure 3.3, however, in the following this energy will be neglected.

Given the power model discussed above, let us extend the half duplex energy model from Section 2.1. We are looking at a star topology with sensor nodes who are transmitting uplink data only. To recap, the half duplex energy consumption per correctly received bit was defined as

$$\bar{\mathcal{E}}_{\mathrm{b}}^{\mathrm{HD}} = P_{\mathrm{TX}} T_{\mathrm{b}} \left(\bar{\tau}_{\mathrm{d}} + \bar{\rho}_{\mathrm{i}} + \bar{\rho}_{\mathrm{c}}^{\mathrm{HD}} \right), \tag{3.2}$$

where the time per bit (T_{b}) is defined as the sum of the transmit time of the payload (T_{P}), the transmit time of the header (T_{H}) and the transmit time of the overhead for synchronization (T_{O}), divided by the number of bits in the payload (L_{P}), i.e.,

$$T_{\mathrm{b}} = \frac{T_{\mathrm{P}} + T_{\mathrm{H}} + T_{\mathrm{O}}}{L_{\mathrm{P}}} \quad . \tag{3.3}$$

For the collision detection, let us assume it takes at least T_{H} and T_{O} to detect them, as a collision always occurs at the beginning of the packet transmission. The time per bit to detect interference can then be defined as

$$T_{\mathrm{c}} = \frac{T_{\mathrm{H}} + T_{\mathrm{O}}}{L_{\mathrm{P}}} \quad . \tag{3.4}$$

Interference on the other hand can occur at any moment during the packet transmission. Therefore, on average the time per bit to detect a collision can

be defined as

$$T_{\mathrm{i}} = 0.5T_{\mathrm{b}} \ . \tag{3.5}$$

Using the above timings we can define the average energy per bit of a transmission using full duplex ($\bar{\mathcal{E}}_{\mathrm{b}}^{\mathrm{FD}}$) as,

$$\bar{\mathcal{E}}_{\mathrm{b}}^{\mathrm{FD}} = P_{\mathrm{RXTX}} \left(T_{\mathrm{b}} \bar{\tau}_{\mathrm{d}} + T_{\mathrm{i}} \bar{\rho}_{\mathrm{i}} + T_{\mathrm{c}} \bar{\rho}_{\mathrm{c}}^{\mathrm{FD}} \right) . \tag{3.6}$$

By definition, $T_{\mathrm{c}}, T_{\mathrm{i}} << T_{\mathrm{b}}$ and therefore the term between brackets in (3.6) will be much smaller than the one in (3.2). However, as can be seen from Table 3.1, $P_{\mathrm{RXTX}} > P_{\mathrm{TX}}$. In a scenario with no interference ($\bar{\rho}_{\mathrm{i}} = 0$) and no collisions ($\bar{\rho}_{\mathrm{c}} = 0$), it is clear that $\bar{\mathcal{E}}_{\mathrm{b}}^{\mathrm{FD}} > \bar{\mathcal{E}}_{\mathrm{b}}^{\mathrm{HD}}$. With interference and collisions, this is no longer clear. In the next section we will investigate this trade-off numerically.

3.3 Collision detection performance improvements

The results in this section assume a star topology where all nodes send traffic in the uplink to the central node. Moreover, all nodes use the CSMA/CA or FD-CSMA/CD protocol to access the medium. First, let us look at the delay before moving to the retransmissions and energy consumption. The delay results come from our MATLAB simulator documented in [30], while the retransmissions and energy results come from [17]. Comparing half duplex with collision detection in terms of delay, we see that for a low number of nodes there is no benefit of using one or the other (Figure 3.4). This is because no retransmissions are necessary as collisions are low. For larger networks, it can be seen that the delay in a half duplex network starts to increase faster than in the network with collision detection.

To evaluate the energy performance we first need to look into the average number of retransmissions. The following evaluations were performed using parameters from the IEEE 802.15.4 and IEEE 802.11 standards, and assuming that the network is saturated. Furthermore, a 25 dB SNR link is considered, which results in $\bar{\tau}_{\mathrm{d}} = 2.2575$ for IEEE 802.15.4 and $\bar{\tau}_{\mathrm{d}} = 1.0101$ for IEEE 802.11. For the collision probabilities, we numerically evaluated the model presented in [18] using the corresponding values for $CW_{\min}$ and $CW_{\max}$. In Table 3.2, all parameters used in the rest of this section are summarized.

The total number of transmission trials, i.e., $\bar{\tau} = \bar{\tau}_{\mathrm{d}} + \bar{\rho}_{\mathrm{i}} + \bar{\rho}_{\mathrm{c}}$, are shown in Figure 3.5 as a function of the number of nodes in the network. Collisions are a bigger problem in IEEE 802.15.4 than in IEEE 802.11, which is to be expected as the backoff period is much smaller in the former, cf. Table 3.2. Moreover, as

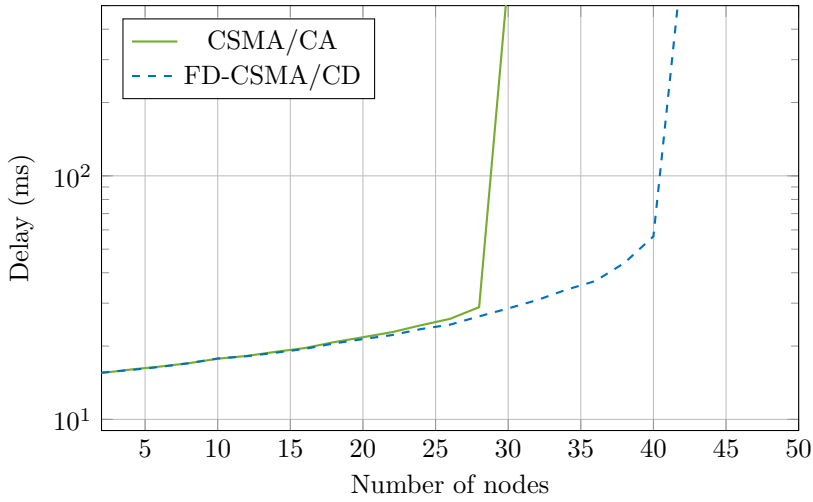


Figure 3.4: Delay analysis using a fixed throughput of 3 packets/s with 10% of the packets downlink. (Source: [30])

Table 3.2: Parameters used for numerical evaluations

Parameter	IEEE 802.15.4	IEEE 802.11
Frame Header — T_{H}	416 μs [‖]	10.25 ns[Φ]
Payload — T_{P}	4.064 ms[‖]	209.48 ns[Φ]
Overhead — T_{O}	128 μs [‖]	3.69 ns[Φ]
Payload length — L_{P}	127 bytes[‖]	1023 bytes[Φ]
TX power — P_{TX}	30.67 mW[*]	824.4 mW[§]
RXTX power — P_{RXTX}	56.95 mW[*]	982.62 mW[§]
Min. contention window — CW_{min}	8[‖]	32[Φ]
Max. contention window — CW_{max}	32[‖]	128[Φ]

From datasheet of: [*]TI CC2420, [§]TI CC3200. Source: [‖][12], [Φ][13] .

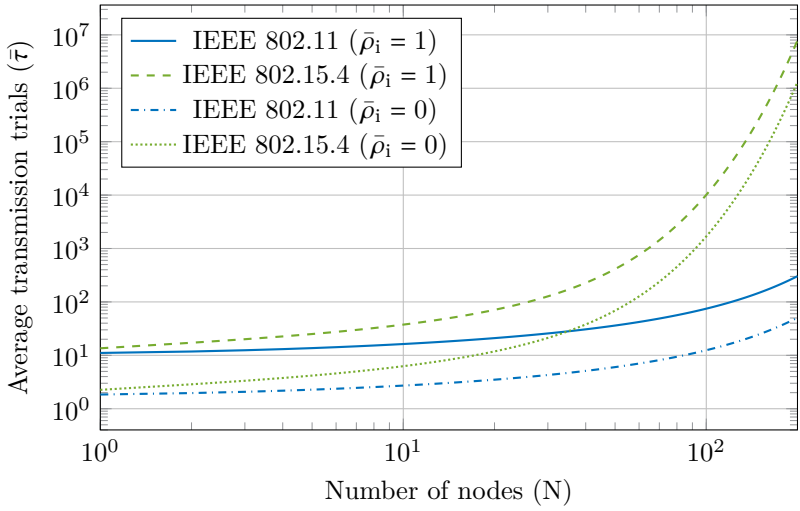


Figure 3.5: The comparison of the number of transmission trials between IEEE 802.15.4 and IEEE 802.11 shows that collisions are a bigger problem in IEEE 802.15.4. (Source: [17])

$\bar{\tau}_d$ and $\bar{\rho}_i$ are independent of the number of nodes, an increase only translates the curve in the horizontal direction.

Using the results from Figure 3.5, it is possible to evaluate (3.2) and (3.6) and compare the two. In Figure 3.6, only the results for the IEEE 802.11 standard are shown but similar results can be found for the IEEE 802.15.4 standard. Let us look at the curves for $\bar{\rho}_i = 0$, i.e., no interference. It is clear that due to the increased power cost of the transceiver and the minimal amount of collisions, the energy consumption of collision detection is higher. However, when we increase the number of nodes, and consequently the amount of collisions, the two lines cross and ultimately the energy consumption is only one 10th of the half duplex energy performance. If we increase the interference, it is a slightly different story. Now even in small networks half duplex performs worse in terms of energy consumption than collision detection. These energy results really show the strength of collision and interference detection, as the time wasted is significantly decreased and therefore the system becomes more energy efficient.

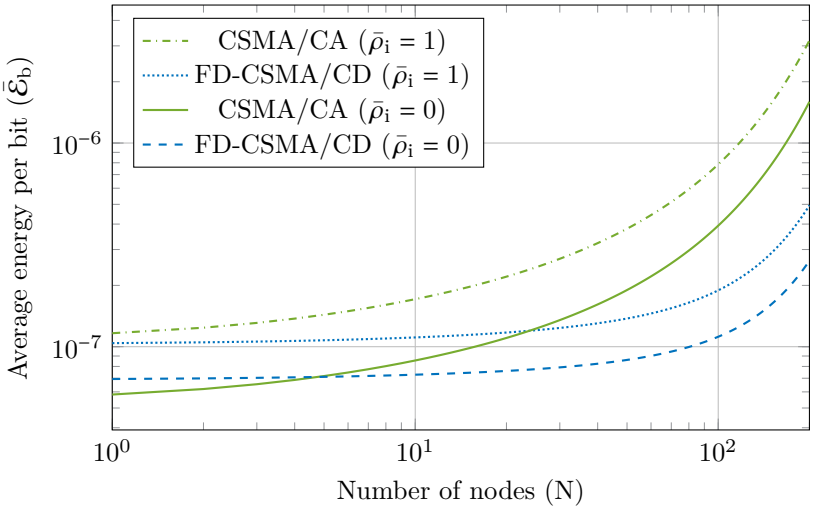


Figure 3.6: Energy consumption of a wireless node per correctly transmitted bit, evaluated using (3.2) and (3.6) and considering parameters corresponding to the IEEE 802.11 standard. (Source: [17])

3.4 An adaptive switching algorithm

From the previous results, it is clear that in small networks or networks with a small number of active nodes, half duplex communications are more energy efficient, while in bigger networks or networks with external interference the collision detection scheme performs better. Therefore, devices should decide at runtime if they want to enable collision detection or not. This can be done by determining the number of active devices in the network N and the amount of interference q_i. However, at runtime this is practically infeasible as it is difficult to estimate the number of active nodes in a distributed way. Visually, the ideal switching boundary is shown in Figure 3.7. Based on q_c, which is a function of N, and q_i it is possible to decide which MAC protocol is best in the current network situation. The complete derivation of this figure can be found in [33]. Note that networks often have high q_c instead of high q_i. If devices can both estimate q_c and q_i, it is possible to make the perfect decision. However, at runtime it is difficult to differentiate between collisions and interference.

In Algorithm 1, a distributed algorithm is proposed which estimates the congestion and interference, i.e. sum of q_c and q_i. This is done by counting the transmitted *(line 19)* and acknowledged packets *(line 22-28)*. If in the last W packets more than ζ are delivered corrected *(line 13-18)*, the algorithm chooses

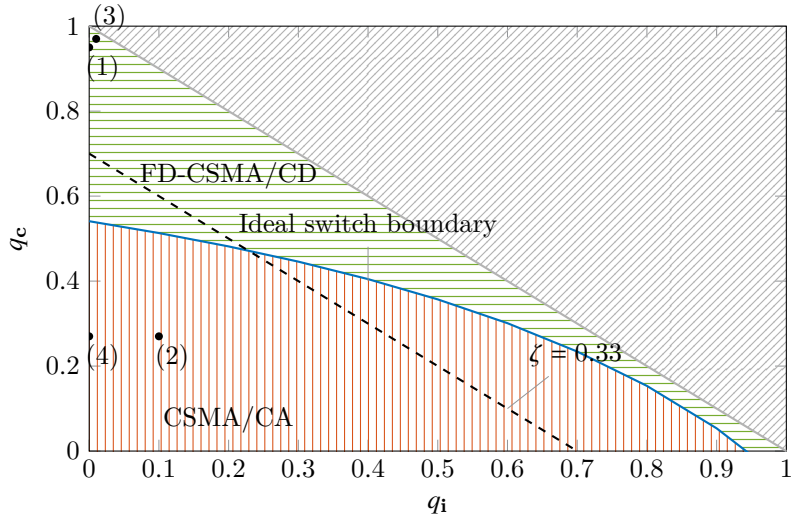


Figure 3.7: The ideal switching boundary (——), splits the area where FD-CSMA/CD is optimal (▭) and the area where CSMA/CA is optimal (▭). The ▨ area is mathematically impossible as the packet error rate is greater than 1. In our practical algorithm $\zeta = 0.33$ (- - -) is used as switching boundary. The four dots mark the four scenarios used in the simulations. (Source: [33])

the half duplex scheme, otherwise it chooses the collision detection scheme. W can be adjusted to avoid excessive switching and acts as a sliding window. ζ corresponds to the packet delivery ratio threshold and is visualized in Figure 3.7. Because we look at the sum of q_c and q_i, we have to make a trade-off between good switching in the low interference region or in the high interference region. In the following ζ is fixed to 0.33 as it is close to the ideal boundary for low q_i. On the other hand a high q_i, i.e. $q_i > 0.5$, typically only occurs with active jammers.

Let us consider a smart home scenario with IEEE 802.15.4 low power sensors to monitor some environmental parameters. The scenario, which was simulated in ns-3, is shown in Table 3.3. In the morning, people walk around the house and generate data but don't use the WiFi. In the afternoon, less people are around so less sensors are activated but the WiFi is used slightly. In the evening, the WiFi is used heavily as well as all the sensors. Finally, during the night, the house is quiet, no WiFi is used and most sensors are turned off. The corresponding q_c and q_i of the different scenarios are shown in Figure 3.7.

In our simulation, the WiFi is modeled as a duty cycled pattern with a period

Algorithm 1 Adaptive selection algorithm

Require: Window $W > 0$
Require: Threshold $\zeta > 0$ **and** $\zeta < 1$
Initialize: $T = \text{zeros}(W, 1)$
function main()
while AvailablePacket() **do**
  free ← DoCCA()
  if free **then**
    TransmitPacket()
    WaitForAck()
  end if
end while
end function

function TransmitPacket()
$C \leftarrow \text{average}(T_{i-W} : T_i)$
if $C > \zeta$ **then**
  Transmit with CSMA/CA
else
  Transmit with FD-CSMA/CD
end if
i++
end function

functionWaitForAck()
if Ack timer **not** timeout **then**
  $T_i \leftarrow 1$ # Packet arrived successfully
else
  $T_i \leftarrow 0$ # Packet did not arrive
end if
end function

Table 3.3: Scenarios used in ns-3 simulations.

Scenario	1	2	3	4
Time of day (6h)	Morning	Afternoon	Evening	Night
number of sensors	100	10	100	10
WiFi duty cycle	0%	5%	50%	0%
Crossing point	25	24	14	25

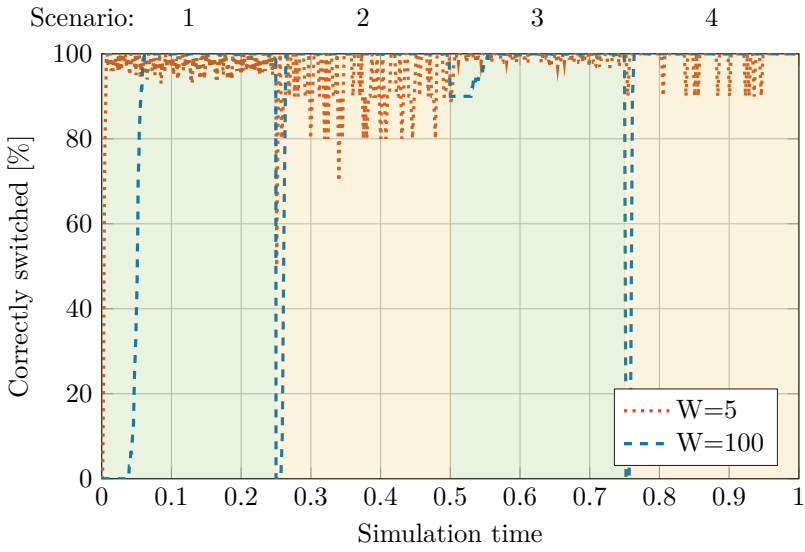


Figure 3.8: The amount of nodes that switched to the correct scheme, as shown in Tab. 3.3. This figure clearly shows that a window size of 5 does not converge to the correct solution. (Source: [33])

of 30 ms. Therefore, a 50% dutycycle means the WiFi interference is turned off and on every 15 ms. For the sensors, we load their buffers with enough data to ensure that every node always has new data to transmit. Each sensor is connected to the central gateway in a star topology.

The crossing point between the CSMA/CA and FD-CSMA/CD curves can be calculated using the derivations from [33] for different WiFi duty cycles (q_i). From Table 3.3, is it clear that in scenarios 1 and 3, FD-CSMA/CD is more energy efficient. In scenarios 2 and 4, CSMA/CA is more energy efficient.

The algorithm presented here finds an optimal solution at runtime based on the ζ line. By adjusting the window W, it is possible to trade off tracking speed for steadiness. The effect of W is shown in Figure 3.8. A smaller window is able to respond faster but fails to always select the optimal protocol, while a larger window converges better but tracks changes much slower.

Figure 3.9 shows the energy saved compared to CSMA/CA. In scenarios 1 and 3, 50% less energy is consumed by using collision detection. In less dense scenarios (i.e. 2 and 4), the energy loss is less than 50%. Averaging over the four scenarios, static FD-CSMA/CD saves 14% compared to a static CSMA/CA. When the nodes adaptively pick their preferred protocol, the energy savings depend on

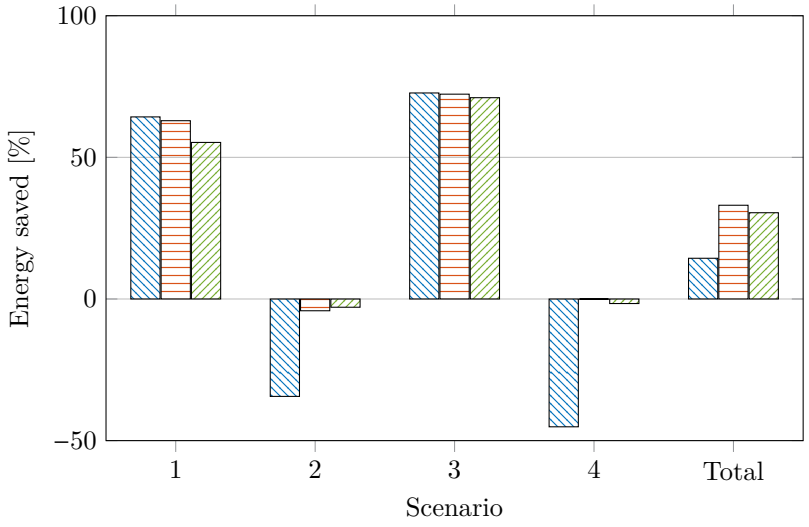


Figure 3.9: The average energy saved for FD-CSMA/CD (▨), W=5 (▭), W=100 (▨). The graph shows clearly that a scheme that adaptively enables the full-duplex collision detection ensures lower power. For scenario 1 and 3, the optimal scheme is FD-CSMA/CD, for the other scenarios, that is CSMA/CA. (Source: [33])

the chosen window. For the small window, 33% of the energy was saved while for the larger window this is 30% for the simulated scenarios.

3.5 Conclusion

In this chapter we have shown the performance gains of detecting collisions and interference during a transmission. In terms of delay, the network collapse can be extended and therefore the average delay is much lower. But the biggest gains can be found in energy efficiency. For networks with a low number of nodes and no interference, detecting collisions only wastes a small amount of power. However, with interference and/or in networks with more than 10 nodes, a 10-fold of the energy can be saved by enabling collision detection. The preliminary results of this chapter are more thoroughly analyzed in Chapter 7.

Moreover, a practical switching algorithm for dense wireless sensor networks is described that dynamically selects the collision detection protocol if high interference is experienced. The tunable window parameter in the algorithm

allows nodes to respond faster to changes in either network usage or interference. The simulated scenarios show the proposed algorithm saves 33% or 17% in energy consumption compared to CSMA/CA or FD-CSMA/CD, respectively.

Now that we have shown the need for detecting collisions through simulations, a logical next step would be to measure the performance in the real-world. Unfortunately, there are no off-the-shelf transceivers capable of doing this. We therefore need to investigate the possibilities to transmit and receive at the same time, as well as focus on prototyping solutions for innovative PHY/MAC technologies. This will be detailed in the next chapters.

Chapter 4

Enabling simultaneous transmit and sense

To detect collisions and interference, a wireless device should be able to transmit and sense at the same time, which is currently not possible. However, recently a new wireless paradigm called in-band full duplex was presented which allows wireless devices to simultaneously use their transmitter and receiver on the same frequency. In this chapter, the problems of enabling in-band full duplex will be detailed, followed by an overview of the state-of-the-art solutions, as well as current use-cases of the technology.

4.1 The self-interference problem

In the past, wireless devices were assumed to be unable to simultaneously transmit and receive on the same frequency due to the strong interference from the self-transmitted signal [34]. In this context, the self-transmitted signal is called the self-interference (SI). To illustrate the SI problem, let us look at the system model in Figure 4.1. Each of the two devices simultaneously transmits a signal, respectively, X_1 and X_2. Part of these signals go to the antenna but another part leaks into the receiver chain. The received signal at device 1 can therefore be defined as,

$$Y_1 = h_{11}X_1 + h_{21}X_2 + N, \tag{4.1}$$

where h_{11} is the channel of the self-interference, h_{21} is the over-the-air channel between device 2 and device 1 and N is a sum of all noise added in the

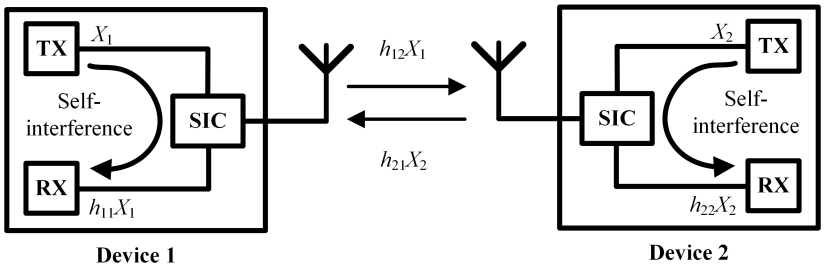


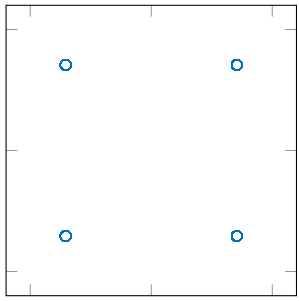
Figure 4.1: The overview of the signals in an in-band full duplex system shows the self-interference problem.

system. Typically, $h_{11} >> h_{21}$ as the path length of the h_{11} channel is orders of magnitude smaller than the path length of h_{21}. In WiFi systems this difference can be more than 100 dB [35].

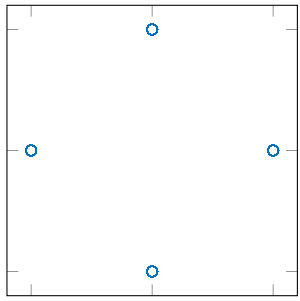
This large difference means that the device won't be able to receive X_2. To give an idea of how these signal would look like, two quadrature phase-shift keying (QPSK) signals were simulated, one for device 1 and one for device 2, as shown in Figure 4.2. The signal from device 1 (X_1) is shown in Figure 4.2a, the signal from device 2 is rotated 45 degrees to clearly see the difference and is shown in Figure 4.2b. If we assume $h_{11} = -10$ dB and $h_{21} = -50$ dB, Figure 4.2c shows the resulting received signal (Y_1) at device 1. It is clear that the incoming signal is dominated by the SI signal from device 1.

Besides this dominating SI signal, the receiver chain is not designed for such strong signals. At the receiver, the incoming signal is amplified using a low noise amplifier (LNA) before it is mixed down to baseband, and digitized by an analog-to-digital convertor (ADC), cfr. Figure 4.4. The LNA is not designed for these strong signals, meaning that it will be driven into its saturation region and completely distort the incoming signal. Now assuming we could design such a LNA, we would need to design an ADC with a dynamic range of more than 100 dB to not only sample the self-interference but also the useful incoming signal with enough bits. Unfortunately, such devices are difficult and expensive to produce.

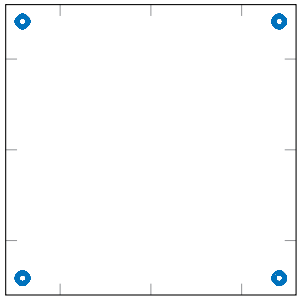
However, in theory we could exploit our knowledge of X_1 and cancel this signal before it reaches the LNA and ADC, allowing us to reuse the current architecture. In practice, our knowledge of X_1 is limited to the digital samples before they are sent to the digital-to-analog convertor (DAC). In [11], the distortions of the transmitter chain were measured, the results are shown in Figure 4.3. On the left side of Figure 4.3 is the signal we are sending to the DAC. It consists of two



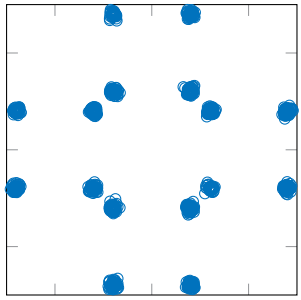
(a) Self-interference signal (X_1)



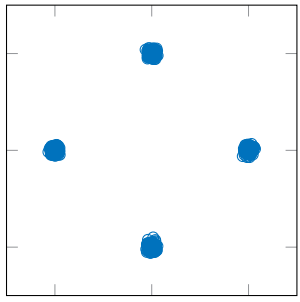
(b) Useful signal (X_2)



(c) Incoming signal before analog cancellation



(d) Incoming signal after analog cancellation



(e) Incoming signal after digital cancellation

Figure 4.2: Due to the strong self-interference signal, it is impossible to detect the useful signal without proper cancellation (c). When analog cancellation is applied (d), already part of the useful signal can be distinguished, while after both analog and digital cancellation (e) we can see the useful signal without much distortions.

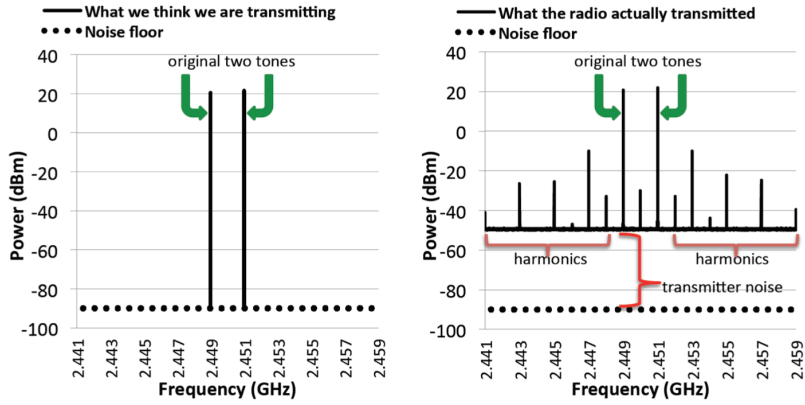


Figure 4.3: What we think we are transmitting from a digital point of view is very different from what the radio actually transmits. The non-linear distortions and transmitter noise are added to the ideal digital signal. (Source: [11])

tones with a spacing of 2 MHz. On the right side are the signals the transmitter is actually transmitting, with all its distortions. These distortions can be split into three categories:

1. **Linear components:** These distortions consist of a linear combination of the two original terms, they can be attenuated and delayed in time.

2. **Non-linear components:** These distortions are due to a non-linear combination of the two original terms. Typically these are caused by intermodulation distortion (IMD), which comes from intermodulation between all the input frequencies and results in additional output frequencies being generated.

3. **Transmitter noise:** These distortions raise the overall noise of the transmitted signal and are rated by the noise figure (NF) of the transmitter. In a typical application, the NF of the transmitter is of lesser importance as the attenuation in the channel will lower this noise below the noise floor of the receiver. However, in IBFD operation, the path length is so short that it becomes a problem.

From Figure 4.3 it is clear that to make IBFD possible, all these components should be canceled to a certain degree. Ideally, we should be left with a SI-free receiver path, meaning that all SI components should be canceled under the noise floor. Using Figure 4.3 we can therefore conclude that in this case the

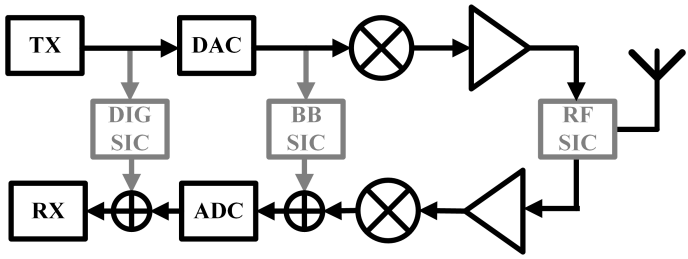


Figure 4.4: A typical in-band full duplex transceiver has multiple stages of self-interference cancellation. The analog stages are crucial to relax the requirements of the LNA and ADC.

linear components should be canceled at least $20 - (-90) = 110$ dB, the non-linear components should be canceled at least $-10 - (-90) = 80$ dB and the transmitter noise, $-50 - (-90) = 40$ dB. Fortunately, not all these components should be canceled before the LNA. As long as the LNA does not saturate, sufficient cancellation is provided. In practice, with a receiver saturation point of -30 dBm, this would mean that around 60 dB of cancellation before the LNA and ADC and around 50 dB of cancellation after the ADC is needed.

4.2 State-of-the-art SI cancellation techniques

From the previous section it is clear that multiple stages of cancellation are needed. They can be categorized into digital self-interference cancellation (SIC) and analog SIC. The latter is split into cancellation techniques at RF and at baseband (BB), as shown in Figure 4.4. RF SIC is the most crucial for the LNA, as the cancellation ensures that the LNA does not saturate. The combination of RF and BB self-interference cancellation ensure enough dynamic range on the ADC. Finally, digital SIC cleans up any remaining SI. The different steps are visualized in Figure 4.2. Before analog cancellation (Figure 4.2c), only the SI is visible, after analog cancellation (Figure 4.2d), the useful signal becomes visible, although heavily distorted. Finally, after digital cancellation (Figure 4.2e) the useful signal is clearly visible.

4.2.1 Analog self-interference cancellation

Several research groups have proven the feasibility of analog SIC. Let us first go over the techniques at RF. These techniques typically focus on reducing the

main component of the SI signal (cfr. Figure 4.3). If more than one antenna is used, this can be achieved by separating the antennas in such a way that the two transmit antennas create destructive interference at the receiver antenna [35]. If precisely placed, this allows to cancel the SI up to 30 dB. However, this only works for a specific frequency and antennas have to be moved to use another frequency. Another technique which uses antennas, is to use dual polarizing antennas [36]. One polarization is used for the transmitter and the other is used for the receiver. The isolation between the two polarizations can be up to 40 dB [36]. The main drawback is that if two devices communicate with each other, their polarization need to be aligned for best SNR.

Before going to techniques using custom hardware, there is one technique which uses existing hardware in a wireless transceiver to cancel the SI. By using an extra transmitter chain it is possible to inject an inverse signal of the SI which destructively interferes with the leaked self-transmitted signal [37]. The idea is to pre-compensate for the distortions in the first transmit chain. Of course the second transmit chain adds distortions of its own which should also be calibrated for. In [37] it is shown that such a technique can achieve a cancellation of up to 23 dB. However, adding a second transmitter chain requires more space on the chip as well as more power. Ultimately the benefits would probably be lower than using the two transmitter chains in a multiple input multiple output (MIMO) configuration.

The techniques that follow in the rest of this section all use custom hardware designed for IBFD operation. The first one uses an electrical balance duplexer (EBD) [38], of which the concept is shown in Figure 4.5. The EBD creates two signal paths for the signal coming from the power amplifier (PA), one part goes to the antenna and is reflected back towards the LNA, another part goes to the balance network and also gets reflected towards to LNA. The reflection from the antenna is due to mismatch of the antenna and causes the main linear components of the SI signal. To achieve cancellation, the balance network, which consists of tunable capacitor banks, should be set to create an inverse copy of the reflected signal from the antenna port. These two signals will add up before the LNA and will destructively interfere. If correctly tuned, such architectures can achieve up to 70 dB of cancellation over a 5 MHz bandwidth and up to 50 dB over a 20 MHz bandwidth. On top of this, in [38] it was shown that these devices can be made very linear, lowering the non-linear and noise components and therefore relaxing the subsequent cancellation steps. However, tuning of the balance network is key, any changes in the antenna reflection should be matched by changes in the balance network. The speed and accuracy at which this can be achieved determines the performance in practical situations. Moreover, due to the signal split at the PA side, there is a 3 dB insertion loss inherent to this architecture, which should be compensated.

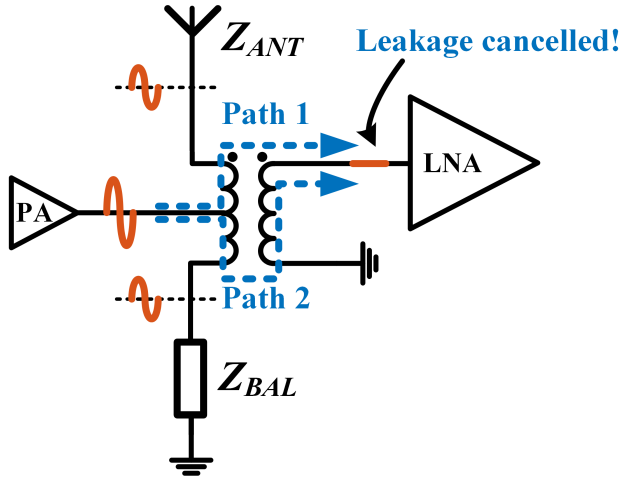


Figure 4.5: The electrical balance duplexer creates two signal paths. At the receiver end, these two signals destructively interfere with each other the cancel to self-interference.

As the linear components of the SI signal are a linear combination of delayed versions of the transmitted signal, it is possible to create a similar path for the cancellation signal. In [11], the signal from the transmit chain is tapped and passed through a network of parallel fixed lines of varying delay and tunable attenuators. These copies are then added up again and subtracted from the received signal. The cancellation is limited by the range of the varying delay lines. Moreover, adding more lines is typically limited by space or power constraints. Still, this technique allows to cancel the SI signal up to 45 dB with careful tuning [11]. The main downside is that this techniques require space on chip for the delay lines, which can be costly.

Finally, at baseband, a vector modulator is a component which can mix a certain signal down to baseband and at the same time apply a variable phase shift and attenuation [39]. The input of the vector modulator is taken at RF while the output is added with the signal coming out of the mixer in the receiver chain. The phase shift and attenuation is adjusted in such a way that both signals destructively interfere with each other. In [39] it is reported that up to 27 dB of cancellation can be achieved. Another vector modulator design taps the signal at the transmitter baseband and then mixes the signal to add a variable delay and attenuation [40]. Their design is able to cancel up to 42 dB, although with very bad linearity specifications. These techniques can be added to all the previous techniques to cancel the SI before the ADC.

4.2.2 Digital self-interference cancellation

In the digital domain, there is more flexibility to cancel the SI signal and clean up any remaining linear and non-linear components. To cancel the linear components, the remaining self-interference channel (h_{11} in Figure 4.1) should be estimated using for example a least squares estimator. The received self-interference can be written in the frequency domain as

$$Y = H_{11}X_1 + N. \tag{4.2}$$

The channel H_{11} can therefore be estimated using

$$\hat{H}_{11} = \frac{Y}{X_1}, \tag{4.3}$$

as Y are just the received samples and X_1 are the known transmitted samples. This estimation can be done on a per-packet basis using the preamble which is available in most wireless standards. Moreover, most of the hardware necessary for these operations is already available in a typical receiver as it is required to compensate for any channel variations.

Unfortunately, the rather simple channel estimation from Equation (4.3) only takes the linear components into account. To cancel the non-linear components, more computations are needed. In [11] a Taylor expansion is used to approximate the non-linear channel. Therefore, the received digital samples can be written as,

$$y(n) = \sum_m x(n)(|x(n)|)^{m-1} * h_m(n), \tag{4.4}$$

where $h_m(n)$ are the terms that need to be estimated. Empirically, in [11] it was found that most higher order (m) terms are zero. This comes from the fact that higher order terms are created by mixing lower order terms together, and therefore the power in the higher order terms reduce. Therefore, in a typical indoor environment only 224 $h_m(n)$ values need to be estimated, which is something that can again be done on the preamble.

After estimating both the linear and non-linear components, the resulting channel is applied on X_1 and subtracted from the received samples, giving

$$Y_d = (H_{11} - \hat{H}_{11})X_1 + N. \tag{4.5}$$

Using the above described techniques another 40 dB can be canceled.

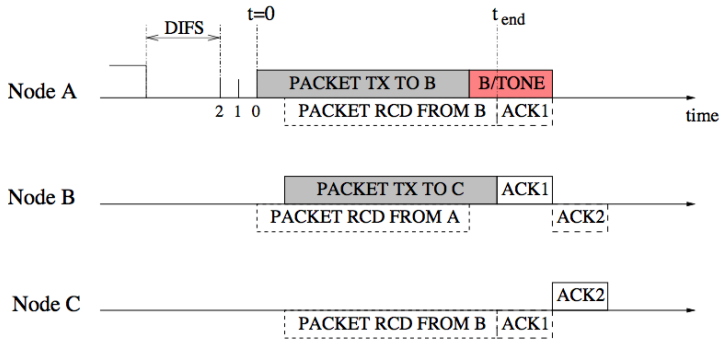


Figure 4.6: The packet transmissions using in-band full duplex relaying show the overlapping transmissions. (Source: [41])

4.3 Current use-cases for in-band full duplex

Most higher layer research concerning in-band full duplex focuses either on symmetric use-cases where two devices simultaneously transmit data to each other, or on relaying use-cases where the relay node simultaneously receives data from one device and transmits this to another device.

An upgraded CSMA/CA algorithm with support for relaying has been detailed in [41]. The protocol is shown in Figure 4.6. Let's assume node A went through the back-off and CCA stages as detailed in Section 2.1 and finds the medium free. At $t = 0$ it decides to transmit a packet to node B. Upon reception of the packet, node B starts its transmission to node C. Based on this transmission, node A knows that node B has received its packet. When the transmission of node A is finished, it will transmit a busy tone to make sure the medium remains occupied and it can, without any collisions, receive the acknowledgment (ACK1) from node B. After ACK1, node C transmits ACK2 to node B. In this case both node A and node B are IBFD-capable. The main benefit of this scheme is that there is an inherent collision notification present on the transmission between node A and node B. However, this collision notification is not present on the transmission between node B and node C. The simulations based on this protocol in [41] show that some throughput gains can be achieved in random access scenarios, although this is very dependent on the direction of the data. If all data flows from node A over node B to node C, then the highest gains are possible. When the data flows randomly, only gains of 50% in terms of throughput are possible.

In the symmetric use-case, the same flow as in Figure 4.6 can be followed

with the only change that node B now transmits back to node A instead of node C. In [42] this protocol was analyzed, and it was shown that if both nodes have enough data to send to each other, the throughput can be almost doubled. However, it is important to note that this is dependent on the SI cancellation. If the self-interference cannot be canceled under the noise floor, the SNR of the secondary transmission (node B to node A) decreases and this will decrease the throughput over the secondary link. The scheme not only increases throughput but also increases access point fairness, a measure of the transmit opportunities of the access point when compared to all nodes in the network. In typical CSMA/CA networks, the access point has the same amount of transmit opportunities as a normal node in the network. However, in networks where the access point has more data to send than the other devices this creates problems. With IBFD and the symmetric protocol, this is completely solved.

4.4 Conclusion

The self-interference problem detailed in this chapter is not easy to solve and therefore requires multiple techniques combined to make it work. To cancel the self-transmitted signal at the receive port, we need a combination of analog and digital techniques, each canceling part of the self-interference below the noise floor. Luckily the feasibility of these techniques has recently been proven. Currently, most research focuses on symmetric in-band full duplex communication where two devices simultaneously transmit and receive data on the same frequency. The concept of transmitting data and simultaneously sensing for collisions has not been prototyped and proven yet. In the next chapter we will go deeper into building a prototype capable of transmitting and sensing at the same time using in-band full duplex.

Chapter 5

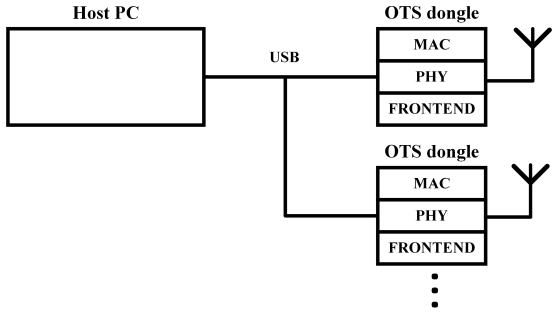
Towards an in-band full duplex prototype

To get a first idea of how a PHY and MAC layer algorithm performs, it is always a good idea to start with simulations or analytical analysis. These allow researchers to get a basic idea of how their algorithms will perform, given the simplifications required for the simulations. In our case, some first results were presented in Section 3. A logical next step is to drop these simplifications and test the algorithms in the real world using some sort of prototyping platform. Typically, this is where the PHY layer and MAC layer communities go their own way, making it difficult to test cross-layer algorithms like the ones in this doctoral thesis. This chapter will provide an overview of the current prototyping platforms as well as a detailed look into the in-band full duplex prototype developed in this PhD.

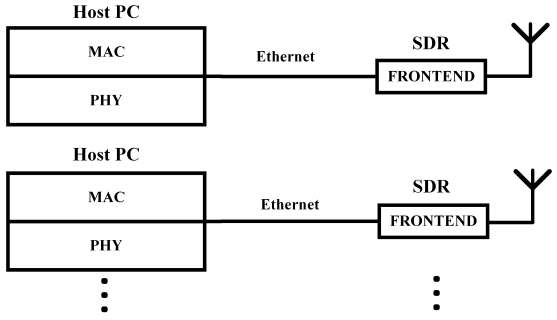
5.1 Overview of prototyping techniques

5.1.1 MAC layer prototyping

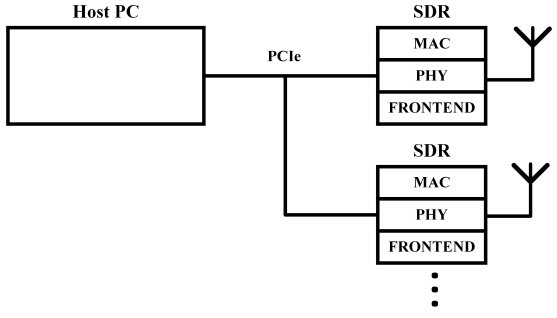
The MAC layer community most often needs a platform which can give them instant response time to test their MAC algorithms. They are interested in benchmarking their algorithms in terms of throughput and delay. To achieve this real-time behavior, some fixed off-the-shelf transceiver dongle is used. These dongles have a small microprocessor on board which allows to run the MAC



(a) Fixed real-time



(b) Flexible non-real-time



(c) Flexible real-time

Figure 5.1: Based on the requirements of the community, different approaches are used for prototyping and benchmarking novel algorithms.

algorithm in real-time. An example setup is shown in Figure 5.1a. A PC is used for controlling the experiment and logging the measurement data. Then using a USB connection, the dongles are connected and programmed. The dongles integrate the MAC, PHY and front-end on chip. The main benefit of this setup is that it allows MAC researchers to validate their algorithms with a minimal effort as only the microprocessor needs to be programmed. On top of this, due to the hardware PHY layer, the experiments can run in real-time.

A typical prototyping platform combines a standard-compliant transceiver with a microprocessor. Here, the transceiver is the fixed part running the PHY and the microprocessor is the flexible part, running the MAC. For IEEE 802.15.4 often the TI CC2520 [43] transceiver is used in conjunction with a low power microcontroller. For IEEE 802.11, the ESP8266 [44] is often used for its low cost. It features a 32-bit microcontroller integrated with an IEEE 802.11 b/g/n transceiver. Similar products exist for other standards as well.

To make it even easier for MAC researchers to develop their solutions, several groups have tried to expose as many features of these transceivers to the user space and make this as flexible as possible. For wireless sensor networks, there exists several operating systems like tinyOS [45] and Contiki [46], which really focus on low power devices. Unfortunately, the development of these operating systems has stopped. OpenWSN [47], on the other hand, is still in active development, and features full support for sensor network standards like, IEEE 802.15.4e and IPv6 over low power wireless personal area networks (6LoWPAN) [48]. OpenWSN is compatible with many of the off-the-shelf sensor nodes. Another approach called snapMAC [49] provides a generic architecture to easily compose and program new MAC design. snapMAC allows to build time-critical algorithms by writing the algorithm as a chain of time-annotated commands. This promises developers to allow them to create new algorithms instantly.

In terms of testbed facilities, the Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT) testbed is one of the bigger ones worldwide. It features around 400 nodes each with a range of radio resources, including WiFi 802.11a/b/g, 802.11n, 802.11ac, Bluetooth and ZigBee. Experimenters can reserve all or part of the testbed and run MAC and network layer experiments in a real environment. For MAC-only experiments, this is great as all the hardware and software is already in place. Another testbed is w-iLab.t [50], which has very similar features as the ORBIT testbed.

5.1.2 Physical layer prototyping

The PHY layer community on the other hand wants maximum flexibility, as their algorithms are not standard compliant. Their main objective is to

improve spectral efficiency or decrease interference to other transmissions. These measurements do not need to be real-time. A typical SDR, like the NI USRP-2922 [51], gives them the flexibility to experiment with new algorithms. All PHY components run on the host PC and only the IQ samples are sent to the SDR. An example setup is shown in Figure 5.1b. The SDR is connected over ethernet in this case to the PC. In this setup, typically multiple PCs are used as it requires a lot of processing power from the PC. The main benefit is again that development and debugging is easy as it can be done on a regular PC. Moreover, the same development environment can be used for simulations and for the actual measurements.

The NI USRP-2922 SDR can be used with GNUradio [52], a software platform with lots of open-source implementations of for example IEEE 802.11 [53] and IEEE 802.15.4 [54]. All these PHY layer implementation run on the host and are therefore not real-time enough for MAC layer experiments. However, FPGA support is slowly being introduced into GNUradio for example through RFNoC [55]. IRIS [56] is a similar modular software defined radio platform that really focuses on runtime flexibility. It has some MAC layer blocks as well, but none of them run in real-time. Another software platform is NI LabVIEW [57] and NI LabVIEW FPGA [58] which is used in this work. LabVIEW features a lot of the building blocks for PHY layer development but as with GNUradio, lacks the MAC layer blocks.

Using GNUradio, a full IEEE 802.15.4 implementation was made running on a host PC and using the universal software radio peripheral (USRP) hardware as frontend [59]. The implementation requires a very powerful computer as the PHY, MAC and network layer all run on the PC. The IQ samples are then streamed over the gigabit ethernet connection to the USRP. In theory, this setup should be able to stream 25 MHz of I and Q samples in both directions, but in practice, the host PC is not fast enough to handle this amount of data. Therefore the bandwidth is often limited at 10 MHz. Due to the interaction with the host, the implementation does not include the full CSMA/CA protocol as it is difficult to meet the timing constraints.

For streaming SDR platforms, a design for slotted MAC implementations is presented in [60], which annotates all samples with a timestamp and control information. With this approach, the latency can already be reduced, however the turnaround time still remains high and it is therefore not possible to make a more dynamic network. Similarly, in [61], a MAC framework is presented, built on top of the IRIS SDR platform. Although they are able to reduce the latency within the requirements of most standards, adding more functionality for cross-layer designs would again increase the latency as it is still running on the host PC and not on the FPGA.

5.1.3 Cross-layer prototyping

So what about cross-layer experiments? The MAC layer approach is not useful as there is no flexibility in the PHY layer and typically also not in the front-end. One approach could be to run both the PHY and MAC layer on the host PC. However, due to the latencies between the SDR and the PC, it is already challenging to run this setup in real-time. Therefore, in this work, we opted for a third approach, shown in Figure 5.1c. Here both the PHY and MAC layer run on a FPGA inside the SDR. This allows us to combine the best of both worlds, not only do we have the real-time support but it is also very flexible. To achieve this low latency, the microprocessor needs to be as close to the PHY layer as possible. This can be done by configuring a Xilinx MicroBlaze softcore [62] on the FPGA or by connecting an ARM processor to the FPGA, like in the Xilinx Zynq product line [63].

In both cases, software support is rather thin at the moment, requiring a lot of effort from the researchers. However, some open real-time implementations exist, like Wireless open-access research platform (WARP) for example. WARP provides a custom hardware platform which consists of a Xilinx Virtex 6 FPGA and two transceivers. The project is tailored for cross-layer research of the IEEE 802.11 standard as all supported software and reference design are made for this standard. Therefore the flexibility is limited, developing an IEEE 802.15.4 transceiver with this platform would require some effort.

National instruments also built some reference designs using their LabVIEW Communications System Design Suite [64]. The NI LabVIEW Communications 802.11 Application Framework [65] features a IEEE 802.11ac standard compliant PHY implementation which runs on the FPGA of a USRP. With the proper licenses it is possible to change the source code. Also the MAC layer runs on the FPGA, which is also standard compliant.

In [66], a real-time IEEE 802.11 PHY and MAC implementation on a FPGA is presented. The code is easily configurable and allows cross-layer optimizations. The FPGA runs the CSMA/CA MAC layer as well as a soft processor which can be used for the higher layers. Our platform has similar functionalities and will be detailed in the next section.

5.2 The in-band full duplex prototype

The prototype developed in this PhD falls into the third category (cf. Figure 5.1), enabling cross-layer research. To the best of our knowledge, it is the first full IBFD prototype capable of forming a network of connected nodes. It

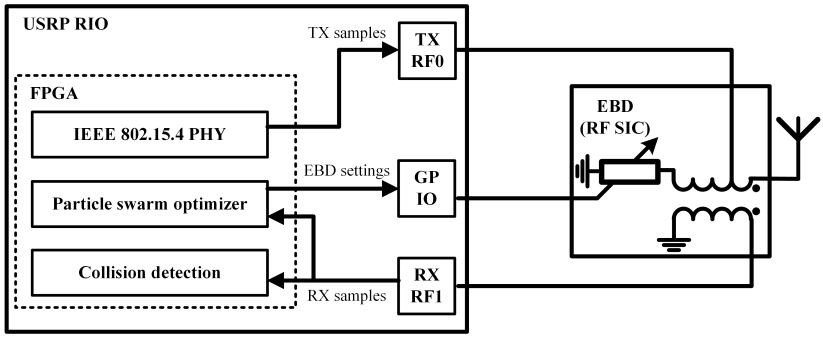


Figure 5.2: Block diagram of the in-band full duplex prototype. (Source: [69])

features a FPGA-based IEEE 802.15.4 PHY and MAC implementation running in real-time on a SDR. In this section, we will go deeper in on the lower layers of the prototype and leave the networking related features to Chapter 10.

5.2.1 Overview of the prototype

The lower layers of the prototype consist of three main parts, as shown in Figure 5.2. All parts are implemented on the FPGA of a USRP SDR. Once the SDR is configured, it can run standalone without interactions from the host PC. This enables us to do real-time benchmarks of our novel collision detection algorithm. The first part, the IEEE 802.15.4 PHY, consists of a flexible implementation of the PHY which is fully compatible with off-the-shelf nodes and can therefore be used to benchmark any new solutions with existing solutions. The implementation, which is detailed in Chapter 9 and published in [67], is split into functional blocks which can be easily upgraded for cross-layer optimizations. The compatibility with off-the-shelf nodes and easy upgradability was published and demoed at MobiCom 2015 [68]. The setup, shown in Figure 5.3, featured three off-the-shelf nodes transmitting at different frequencies to a multi-channel receiver based on our implementation. The USRP is able to run at least four simultaneous transmitter and receiver chains in parallel in real-time. Both the transmitter and receiver chain directly connect to the USRP front-end on the one side and to the MAC layer or host PC on the other side. This allows for easy benchmarking of the PHY layer using the host but also enables embedded real-time experiments using the MAC processor.

Next, we have the analog self-interference cancellation, which uses a particle swarm optimizer (PSO) [70] to find the correct setting. The PSO algorithm will be explained in Section 5.2.2. The implementation was published and demoed

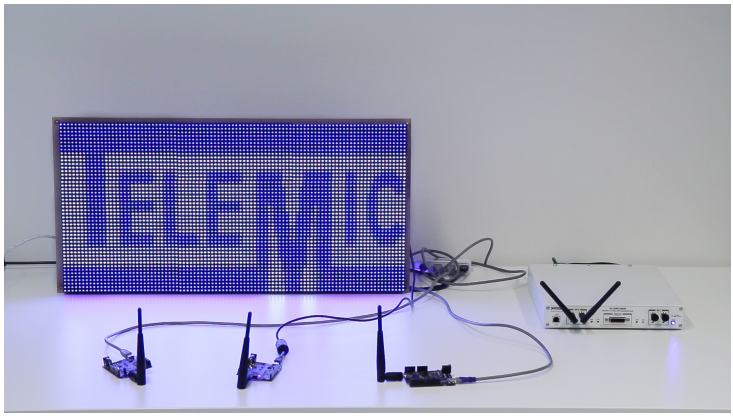


Figure 5.3: Overview of the MobiCom demo setup showing one USRP with our implementation loaded on the FPGA and three off-the-shelf nodes. (Source: [68])

at IEEE DySPAN 2015 [71] and will be detailed in Section 5.2.2. Finally, there is the collision detection algorithm which will be theoretically explained in Chapter 8 and was published in [72]. The implementation of this algorithm was published and demoed at IEEE DySPAN 2017 [69] and will be further explained in Section 5.2.3.

5.2.2 Real-time self-interference cancellation

The prototype focuses on analog self-interference cancellation at RF, as this is the most important cancellation. Digital cancellation is not necessary at this point as our digital collision detection algorithm can handle some remaining self-interference. The hardware setup consists of three parts, as shown in Figure 5.4, (1) an electrical balance duplexer, (2) an interfacing PCB and (3) an SDR for the digital control.

The electrical balance concept was detailed in Figure 4.5. The integrated circuit that we are using was designed by imec [38]. The circuit is produced by using a partially depleted 0.18 μm RF silicon-on-insulator CMOS technology. The design is focused on linearity to keep the non-linear self-interference terms as low as possible. The input-referred third-order intercept point was measured to be over +70 dBm, enabling it to withstand high power self-interference without any significant non-linearities at the receive port. The insertion loss at the transmitter side is less than 3.7 dB and at the receiver port it is less than 3.9 dB. The balance network consists of two fixed inductors, a fixed 50 Ω termination,

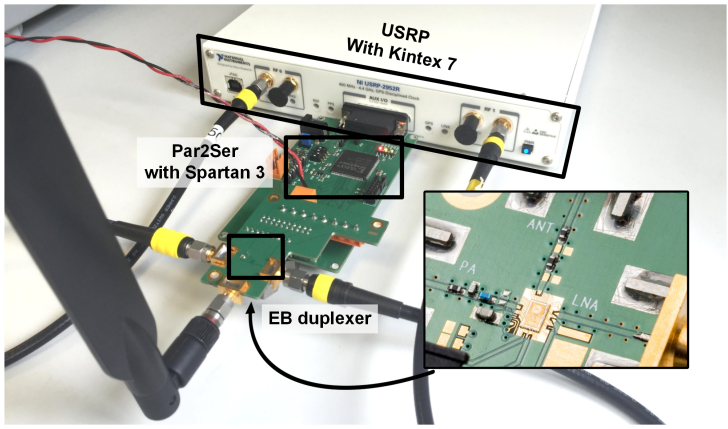


Figure 5.4: Hardware overview showing the USRP, interfacing PCB and EBD. (Source: [71])

and four 8-bit tunable capacitor banks, which enable the balance network to create a perfect inverse-phase copy of the self-interference signal. Unfortunately, because of the high flexibility, the four tunable capacitor banks create a four dimensional optimization space with over 4 billion settings.

At runtime it is not possible to go over all these settings to find the optimal solution. Therefore, a fast optimization algorithm is needed to find a setting which satisfies a certain cancellation level. For this, a Xilinx MicroBlaze softcore is configured on the FPGA of the USRP running at 150 MHz. To find an optimal setting, a PSO [70] is used, which is an optimization method for continuous non-linear functions. A PSO randomly distributes particles into the hyper-dimensional optimization space and gives them random velocities on where to move next. Next, each particle moves and keeps track of its best position and the global best position. At each step of the algorithm, the particles change their velocities based on the personal and global best positions and on a random weight. These steps are repeated until a certain goal is reached. In the EBD case, the position consists of a four dimensional space with on each axis one of the capacitor banks. The fifth dimensional is then the amplitude of the remaining self-interference, which should be minimized. The full PSO algorithm is implemented in C and only uses 20 KB of memory.

To connect the USRP with the EBD, a custom PCB is designed which uses ten general purpose input output (GPIO) pins of the USRP as input. The Xilinx Spartan 3 [73] then converts the parallel inputs to a serial output which can be send to the input pin of the EBD. This circumvents two problems, on the

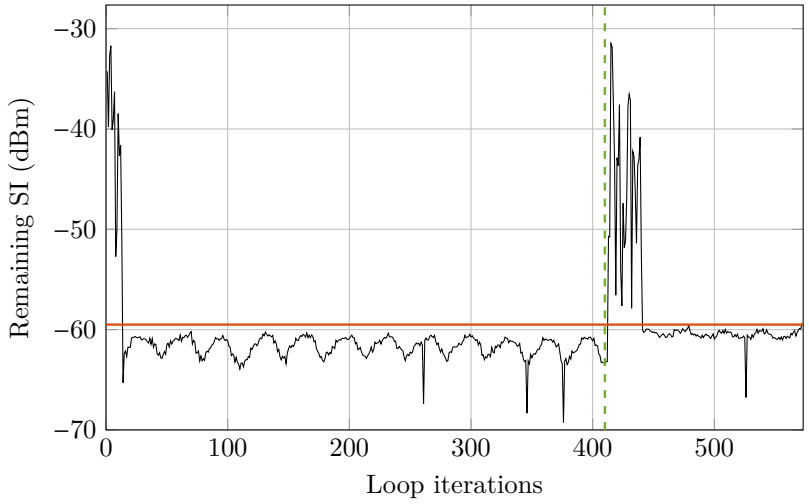


Figure 5.5: Remaining SI after EBD cancellation over time. (——) is the threshold of the PSO algorithm and (- - -) indicates when the antenna impedance changes. (Source: [71])

one hand the GPIO pins of the USRP are too slow, on the order of 10 MHz, at higher frequencies the distortions become too big. The Spartan can therefore increase the speed ten-fold to 100MHz, which is around the limit of the switched capacitors in the balance network. On the other hand, the interfacing PCB creates an easy way to connect the two. Moreover, the output voltage of the USRP does not match the input voltage of the EBD, therefore a level shifter is placed in between as well.

Figure 5.5 shows the performance of the PSO algorithm. It shows the remaining SI at the position of the best particle over time for a transmit power of 0 dBm. The goal of the PSO algorithm is to optimize the remaining SI below -60 dBm and therefore achieve a cancellation of 60 dB. At start-up time, all particles are randomly initialized and move around until a solution below -60 dBm is found. Just over iteration 400, we introduce an antenna impedance change by moving a metal object close to the antenna. This means that the current setting of the EBD is no longer optimal. The PSO software detects this and randomly initializes the particles again to restart the PSO algorithm. On average, it is possible to find a solution around -60 dBm within 1 ms. The main bottleneck being the switching time of the capacitors inside the EBD, as these are optimized for linearity, they have a limited speed.

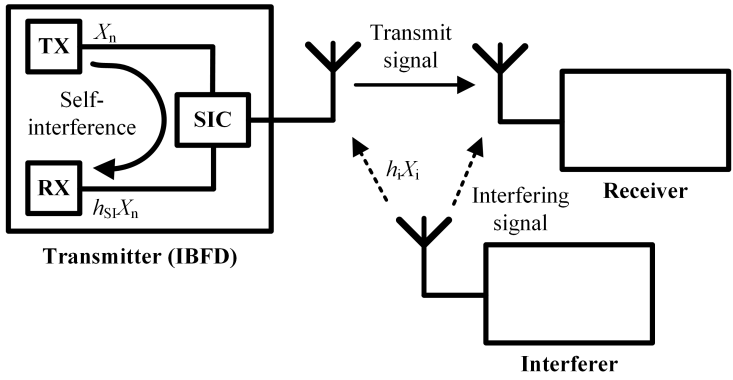


Figure 5.6: Overview of the considered system for detecting collisions and interference. (Source: [69])

5.2.3 Detecting collision in real-time

With the cancellation in place, it is possible to implement the collision detection algorithm. A thorough theoretical background and comparison of the algorithms is provided in Chapter 8, here we will detail the prototype and its performance. Using the system of Figure 5.6, the detection problem can be written as a binary hypothesis,

$$Y_n = \begin{cases} h_{SI}X_n + W_n & \text{if } \mathcal{H}_0 \\ h_{SI}X_n + h_iX_i + W_n & \text{if } \mathcal{H}_1, \end{cases} \tag{5.1}$$

where $h_{SI}X_n$ is the remaining self-interference received by the IBFD transmitter, W_n is the noise, and h_iX_i is the interfering signal received by the IBFD transmitter. The two hypotheses can be defined as $\mathcal{H}_0$ being the case when no collision is occurring and $\mathcal{H}_1$ being the case when there is a collision.

To differentiate between the two hypotheses, we use a goodness-of-fit test called the Kuiper test [74]. A goodness-of-fit test calculates the distance between two empirical cumulative distribution functions (CDF). When this distance is above a certain threshold, the null hypothesis is rejected. In the case of the Kuiper test, this distance is defined as

$$\begin{aligned} T_{KP} = &\sup_{\tau}\{\hat{F}_Y(\tau) - \hat{F}_{Y_0}(\tau)\} \\ &+ \sup_{\tau}\{\hat{F}_{Y_0}(\tau) - \hat{F}_Y(\tau)\} > \lambda_{KP,\alpha}, \end{aligned} \tag{5.2}$$

where $\hat{F}_{Y_0}(\tau)$ is the reference empirical cumulative distribution function (CDF) of the received samples under $\mathcal{H}_0$. These values can be saved when there are no

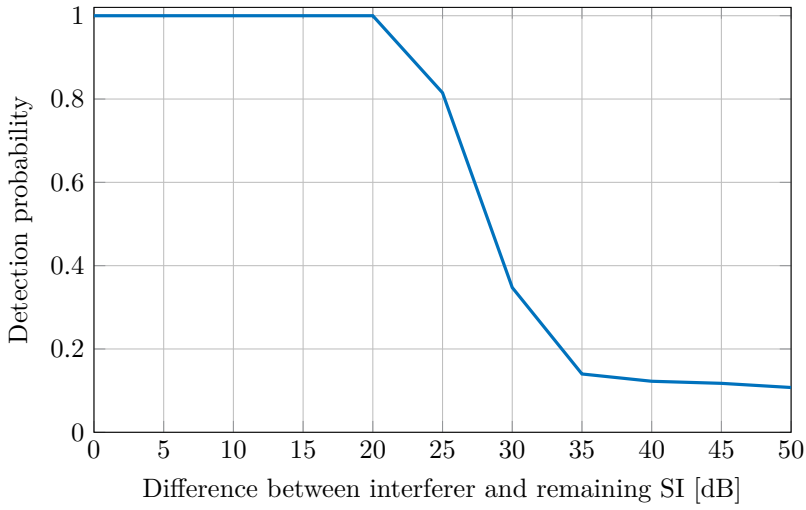


Figure 5.7: Measured detection probability using our collision detection prototype. (Source: [75])

interferers present. $\hat{F}_Y(\tau)$ is the empirical CDF of the current received samples and $\lambda_{\mathrm{KP},\alpha}$ is the threshold which can be trained in advance.

This entire algorithm is implemented on the FPGA of a USRP. Chapter 10 provides a more in depth explanation of the implementation. For now, let us look at the detection performance of the algorithm, shown in Figure 5.7. Given the setup in Figure 5.6, the algorithm is able to detect collisions and interferers with a signal power which is 20 dB below the remaining self-interference with a close to 100% reliability. Throughout these measurements, the false alarm rate was below 5%. This is, to the best of our knowledge, the first real-time implementation able to detect in-band collisions at the transmitter under self-interference.

To put these numbers into perspective, let us assume an analog self-interference cancellation of 60 dB, a transmit power of 0 dBm and a noise floor of -90 dBm. The level diagram in Figure 5.8 shows the range of detectable interferers. This means that any collision or interferer with a power between 0 dBm and -80 dBm can be detected. Increasing the analog cancellation or adding digital cancellation can decrease the lower bound up to the noise floor, as in [72] we have shown that the detection algorithm is limited by the noise floor.

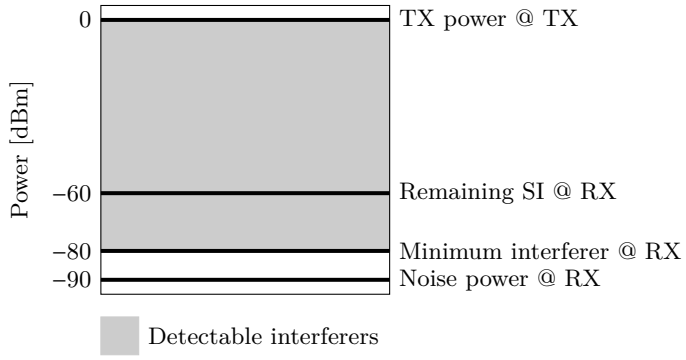


Figure 5.8: All power levels and the detectable power of the interferers for our detection algorithm. (Source: [69])

5.3 Conclusion

This chapter presents an overview of prototyping techniques for PHY and MAC layer research. The PHY layer community typically uses software defined radios as these give them the flexibility needed. The MAC layer community doesn't need this flexibility but requires real-time performance from the prototype to test their protocols in a network of devices. To do cross-layer research with either of these approaches is difficult as for this type of research, flexibility with real-time performance is needed. Some platforms are already starting to offer this but support is still rather thin.

Therefore, in the second part of this chapter, our prototype using in-band full duplex is presented. The prototype is built on a USRP with a big FPGA onboard. The FPGA takes care of all the real-time requirements as well as the flexibility needed to develop novel PHY layer algorithms. Using this approach a prototype was built which uses IBFD to cancel the self-transmitted signal and then uses its receiver chain to detect collisions during a transmission. This novel PHY layer feature is then used by the MAC layer to optimize the channel access performance.

The prototype is used in Chapter 10 to form a network of six IBFD-enabled software defined radios. To the best of our knowledge this is the first time IBFD is tested in a network setting. Moreover, it is the first measurement of collision detection in a wireless network.

Chapter 6

Conclusions and future work

In this final chapter, the doctoral work is summarized as well as some future work is discussed. First, the individual conclusions of the main papers that are bundled in the next chapters are summarized, followed by an overall conclusion of this work. Furthermore, the next steps in terms of future work are presented later in this chapter.

6.1 Paper conclusions

In the second part of this doctoral work, four main papers are presented. The first paper (Chapter 7) concerns a full theoretical analysis of the performance improvements of collision detection using in-band full duplex. The second paper (Chapter 8) investigates and compares techniques to enable collision detection at the transmitter where the received signal is affected by self-interference. The third paper (Chapter 9) details the basis of our prototyping platform which enables cross-layer research. Finally, the fourth paper (Chapter 10) gives a comprehensive look at our full prototype which uses in-band full duplex to enable collision detection. Moreover, the improvements are validated in a network of connected software defined radios. We first summarize and conclude these papers before going to the main conclusions and future work.

6.1.1 Paper 1: Performance Analysis of In-Band Full Duplex Collision Detection

In this paper, presented in Chapter 7, the congestion problem in dense contention-based networks is analyzed. All devices are assumed to be in a star-topology with one central node. To start, we run ns-3 simulations of the half duplex CSMA/CA protocol. This shows that the number of retransmissions exponentially increases with the number of active devices in the network. To solve this problem, we propose a novel MAC protocol which is able to detect collisions early on and abort the transmission. This new protocol is also simulated in ns-3 to compare it with the half duplex case. As expected, the simulations show that the amount of retransmissions does not decrease in a saturated network. However, the duration of each retransmission is reduced significantly. For completeness, we also simulate the symmetric full duplex algorithm, presented in Section 4.3. Similar to the half duplex results, the number of retransmissions does not decrease and retransmissions are transmitted in full.

Besides retransmissions, we can draw some conclusions in terms of throughput from our ns-3 simulations. In symmetric traffic conditions, i.e., when the central node always has a packet to send back to the sensor node, symmetric full duplex has the highest throughput. It almost achieves a factor two increase in throughput irrespective of the number of nodes in the network. The throughput gains for collision detection are small for a low number of nodes, around 12% increase for 10 nodes. However, for dense networks with for example 200 nodes, the throughput is increased more than two-fold to 120%. In asymmetric traffic conditions where the central node has only 10% of the total amount of packets to sent back to the nodes, the benefit of symmetric full duplex almost completely disappears. However, the benefit of collision detection remains the same.

To compare the three MAC protocols in terms of energy consumption, we propose a novel energy model which takes the extra energy consumption of in-band full duplex into account. All cancellation stages have an energy number associated with them which is based on empirical knowledge of the device. Using this energy model and the simulation results from ns-3 it is again possible to compare the three protocols in terms of energy consumption. Interestingly, the energy per successfully transmitted bit is the same for half duplex as for symmetric full duplex. This means that although the throughput is double, the energy consumption is also double. For collision detection this is where the true benefit can be found. For a low number of nodes, the protocol performs slightly worse than half duplex because there aren't many collisions but the nodes are still using energy to detect them. For a high number of nodes, the difference in energy is large, collision detection consumes 80% less energy for the same number of nodes. Moreover, collision detection allows 45% more nodes in the

network without increasing the energy consumption.

6.1.2 Paper 2: Towards Instantaneous Collision and Interference Detection using In-Band Full Duplex

The main objective of this paper, presented in Chapter 8, is to examine and compare collision detection techniques in the presence of self-interference. But first, the paper presents an analysis of transmitter-based collision detection in terms of detection and false alarm probability. This is important because the collision occurs at the receiver but we are trying to detect it at the transmitter. The analysis shows that at short distances, i.e., less than 5 m, no significant loss in detection probability is found. Even at longer distances of up to 20 m, the overall detection probability is still 80%. Of course if transmitter and receiver move further away, the detection probability ultimately becomes close to zero at large distances.

Next, the energy detector is compared against three goodness-of-fit tests, the Kolmogorov-Smirnov test, the Anderson-Darling test and the Kuiper test. The reason why we chose these tests is because they do not require any prior knowledge of the collision and interference that we are trying to detect. This is important as wireless technologies change rapidly and therefore specific techniques would become out-of-date soon. The energy detector, which just averages out the energy of the received samples, is there as a baseline comparison, as it is the easiest to implement but was already shown to perform subpar. The goodness-of-fit tests first calculate the empirical CDF of the current received samples and compares this with the known distribution of the self-interference. If the distance between the two CDFs, as defined by the three tests, is larger than a certain threshold, it is classified as a collision.

The paper then analyzes the required number of samples needed for the detector to achieve at least 95% detection probability and only 5% false alarm probability. The results are based on measured signals from our IBFD prototype. It is clear that the energy detector requires much more samples than the goodness-of-fit tests in order to achieve the same detection probability. For the goodness-of-fit tests, the number of samples starts to go up exponentially if interferers lower than 25 dB below the self-interference need to be detected. Looking at the detection probability when all detectors use 2000 samples, the Kuiper test comes out on top. It performs around 3 dB better than the Kolmogorov-Smirnov test and around 2 dB better than the Anderson-Darling test. Finally, we also look into the difference in detection probability when other modulation schemes are used. We show that our tests perform exactly the same for all modulation schemes, and therefore are independent of the interferer we are trying to detect.

6.1.3 Paper 3: CLAWS: Cross-Layer Adaptable Wireless System enabling full cross-layer experimentation on real-time software defined 802.15.4

In Chapter 9, our baseline implementation of an IEEE 802.15.4 sensor node is presented. During the implementation, we focus on flexibility and reconfigurability and this is also detailed in the paper. The architecture has four main components, (1) the IEEE 802.15.4 PHY layer, (2) the IEEE 802.15.4 MAC layer, (3) the Contiki OS based network layer and (4) the host-side configuration and experimentation application. The first three are all embedded on the FPGA of a USRP SDR. Let us now go over these four components.

The PHY layer is split into several functional blocks to make it easier to upgrade or reconfigure the PHY. All parameters can be configured from the host at runtime. These parameters include the sampling rate, pulse shape and center frequency for example. Other parameters like the chipping sequence and modulation type can be easily configured offline and then compiled and loaded onto the FPGA. Both the input of the transmitter as the output of the receiver can be configured to go to the host or to the MAC processor on the FPGA. The MAC processor runs in a small processor on the FPGA and takes care of all the time-critical functionality, like sending the acknowledgment. This processor can be easily reloaded at runtime with a new version of the software. The MAC processor is connected to a MicroBlaze softcore which runs Contiki OS. The software on the softcore can also be reloaded at runtime. Our off-the-shelf sensor nodes run the same software. It runs the 6LoWPAN network layer, which allows all SDRs to form a network with the off-the-shelf nodes. Finally, there is the host application which configures all the parameters at start-up time over the PCIe interface. Moreover, it is also possible to run the setup in PHY-only mode, to test new PHY layer improvements. This can be done using a transport control protocol/internet prococol (TCP/IP) connection and a command/response protocol. This allows experimenters to set up automated tests both at the transmitter as well as at the receiver. The host application also activates a TAP interface which can be connected to software like wireshark to sniff all traffic.

The performance of our SDR implementation is benchmarked against commercially available hardware implementations of the IEEE 802.15.4 standard. In our case, we use Zigduino transceivers. On the PHY layer, we look at the bit error rate (BER) performance as function of the SNR. In general, we see only a 4 dB performance difference between our implementation and the off-the-shelf node, which is small enough for most higher layer testing. To test the MAC layer, we measure the time required by our system to respond with an acknowledgment after a correct packet reception. The time (8μs) is well below

the time defined by the standard (192μs) and therefore shows how real-time our implementation is. The network layer performance is tested by setting up an internet protocol version 6 (IPv6) network and measuring the delay using internet control message protocol (ICMP) packets. The measured round-trip time of our SDR setup is 1 ms faster than the round-trip time of the Zigduino, again showing the real-time performance.

To showcase the cross-layer abilities of our implementation, the paper is concluded with a cross-layer design experiment. The experiments consists of building a multi-channel relay node which receives on one frequency and simultaneously transmits on another. In the experiment, the PHY layer was adapted to allow the transmitter and receiver to digitally shift their center frequencies. This requires a new functional block, called a numerically controlled oscillator, to be added to the design. This is already a change which cannot be done on an off-the-shelf device. On the MAC layer, the software was changed to allow the simultaneous reception and transmission of data, something that is not defined in the IEEE 802.15.4 standard and therefore not possible with the baseline implementation. Finally, the networking layer is configured to forward any received packets to the transmitter on another frequency. These changes are then benchmarked against the baseline single channel implementation. It is shown that using these cross-layer tweaks the throughput of the relay node can be increased by more than 70%.

6.1.4 Paper 4: Sense and Abort: a Real-Time In-Band Full Duplex Collision Detection Network

The final paper, presented in Chapter 10, dives into our network of collision detection prototypes. The paper first details our sense and abort prototype, which allows to almost instantly detect collisions and abort the ongoing transmission to save time and energy. The prototype consists of three main components, (1) the analog self-interference cancellation circuit to lower the remaining self-interference, (2) the PHY and MAC implementation, and (3) the collision detection algorithm and its connections to the PHY and MAC.

The analog self-interference cancellation uses an electrical balance duplexer to prevent the self-transmitted signal of leaking into the receiver chain. The EBD creates two signal paths which combine at the receive port. One of the signals is a reflection from the antenna port, the other one is a reflection from the balance network. An optimization algorithm ensures that the latter reflection is the inverse of the first reflection. If both are combined, they will destructively interfere and cancel each other. It is possible to achieve between 50 to 70 dB of cancellation using this technique. The PHY and MAC implementation builds

further on the implementation of Chapter 9 and has some upgraded performance in the receiver chain. Finally, the collision detection is an implementation of Chapter 8, which implements the Kuiper test and empirical CDF calculation.

The paper continues with a performance analysis of the collision detection implementation. It is shown that the implemented algorithm is able to detect interferers with a power of 20 dB below the remaining self-interference with a 100% accuracy. Moreover, the real-time performance is also analyzed. On average collisions can be detected within 306 μs after the start of the collision. To put this into perspective, this means that the collision can be detected within 10% of the duration of an IEEE 802.15.4 packet. The majority of this time is spend in calculating the empirical CDF. The paper also shows that the current real-time implementation achieves 72% of the total ideal achievable throughput improvement.

Finally, the paper presents the measurements of a network of connected collision detection nodes. This is the first network of IBFD-enabled SDRs. The performance is compared to a network of half duplex nodes. All nodes are connected in a star topology with one central node and have their packet buffers always filled completely. First, we verify if the collision probabilities of both half duplex and collision detection are equal in saturated traffic conditions, as this was one of the findings from our simulation analysis. From the results it is clear that both match well and follow a typical inverted exponential. Next, the throughput is measured, where each node tries to send as much data as possible to the central node. For the six node setup, the performance improvement is already more than 25%, which is consistent with our simulations. But as shown in Chapter 7, the real throughput improvements come at a higher number of nodes in the network.

6.2 Overall conclusions

In this doctoral work, a solution for the coexistence of devices in dense networks is presented, analyzed and implemented. The problem being solved here occurs when many nodes use the same frequency band to send data to each other. Because of this, the collision avoidance mechanism present in many MAC protocols being used nowadays breaks down. The best way to improve this is to not only avoid collisions but also detect them if they occur. Secondly, interference, between devices of competing technologies, also degrades performance. These technologies often don't hear each others transmissions and therefore transmit over each other. The best solution would be to change all technologies to be compatible with each other. However, many of these

technologies are very different and would require many changes to current standards. Therefore, detecting this interference would be the best feasible solution.

Both collisions as well as interference cause the packet reception to fail, which means the packet needs to be retransmitted. This causes even more congestion in the network as more and more nodes are buffering packets. Packet failures are bad for throughput as the correctly received amount of data decreases. Moreover, the time between transmitting the packet for the first time and correctly receiving the packet on the other side goes up. Finally, with each retransmission some amount of energy needs to be consumed. This is not a problem for devices plugged in to the electricity net but for battery-powered devices this is a huge problem.

This work presents four key contributions, which enable the deployment of dense wireless networks without losing performance. The contributions, which are summarized below, increase throughput, decrease latency and decrease energy consumption at the same time.

1. First, a complete analysis of our novel MAC algorithm based on collision detection using in-band full duplex is presented. This analysis is made possible by our new energy model which not only models the current state of the art but also our new PHY and MAC layer designs. The analysis shows that the overall throughput can be more than doubled by introducing collision detection. Moreover, for the same energy consumption, 45% more nodes can be present in the network.

2. Second, a thorough comparison of collision detection techniques under self-interference is detailed. The techniques enable MAC protocol improvements by detecting collisions nearly instantaneously. We show that the goodness-of-fit test called the Kuiper test performs best and is able to detect collisions with a power which is 25 dB below the remaining self-interference.

3. Third, all improvements are implemented on the FPGA of a software defined radio for real-time measurements. This includes the PHY and MAC layer of the IEEE 802.15.4 standard, used to benchmark the performance against the state of the art, as well as all the control hardware needed for the analog self-interference cancellation, and all the algorithms for the collision detection. It is shown that the baseline performance of the PHY and MAC is on par with the performance of off-the-shelf hardware. Moreover, the implemented collision detection is able to detect collisions and abort the transmission in less than 10% of the typical IEEE 802.15.4 transmission time.

4. Fourth, the implementation is used to validated our performance improvements of the first contribution in a network of IBFD-enabled SDRs. The experiments confirm our simulated results and encourage further exploration of the topic. It is shown that the throughput can be increased 25%, even for a network as small as 6 nodes. Future work is needed to scale up the network even further.

6.3 Future work

The findings in this work encourage future work, both from a theoretical and experimental point of view. This section is therefore split into two parts, first the more theoretical exploration future work, followed by the future work on experimentation.

6.3.1 Theoretical exploration

This work has already thoroughly explored the benefits of collision and interference detection using in-band full duplex. However, most of the exploration is focused on homogeneous networks where each node follows the same MAC protocol. Extending the exploration to more heterogeneous networks can be useful as this resembles many of the current deployment scenarios. Several effects can be useful to be explored, for example, the effect of collision detection and abortion of the current transmission on other networks. Does it also increase the throughput of neighboring networks? What is the fairness between the two networks? What if not all devices in the network are in-band full duplex capable?

The hidden terminal problem is something that collision and interference detection at the receiver can solve. However, we did not yet investigate the performance benefits of such a scheme. Therefore, our current simulations could be extended with hidden terminals, to measure the effect on the throughput and energy consumption. In this work, we assume that receiver-based detection completely solves the problem as the instantaneous acknowledgment reserves the spectrum surrounding the receiver. However, it would be interesting to model this and investigate if there are cases where this assumption is not valid.

Furthermore, an in-depth comparison of transmitter versus receiver based collision detection should be performed through simulations. Both have several benefits which need to be quantified. For transmitter-based collision detection, no feedback channel is needed and the transmitter can instantly take action.

Moreover, the receiver does not need to be in-band full duplex capable or enable its IBFD capabilities. Receiver-based collision detection on the other hand, should have lower false alarm and mis-detection probabilities. However, taking immediate action when a collision is detected is not possible, and a feedback channel is necessary to notify the transmitter of a collision.

Another future research direction are other detection techniques. The techniques presented in this work achieve perfect detection up to a certain interferer power threshold. However, the implementation is very hardware consuming. Optimizing the delay might require more and faster hardware. Therefore, techniques with similar detection probabilities but lower hardware requirements need to be investigated. One promising technique could be deep convolutional neural networks, as detailed in [76].

The benefits of collision detection should also be investigated in other network topologies. In a mesh network for example, receiver-based collision detection ensures that all transmitters surrounding the receiver are silenced and therefore problems like the hidden terminal problem can be avoided. Moreover, it ensures that the link is reserved for the duration of the transmission and no other transmitters in the neighborhood can interrupt it.

Finally, the context information provided by the collision and interference detection can be used to optimize the MAC and network layer. Smart algorithms can be thought of where transmitters for example learn over time that certain network paths work better than others or that there is a periodic interferer present. The context information allows us to differentiate decoding errors from collision errors, while in normal networks the only information present is whether a packet has arrived or not.

6.3.2 Experimentation

The prototype built in this doctoral work provides a strong basis for future research. A possible extension would be the addition of digital self-interference cancellation. Our current prototype uses only analog cancellation. Digital cancellation can further decrease the noise on the received signal and potentially improve the detection probabilities or lower the required number of samples to detect interferers. The prototype is flexible enough to allow for such an extension. Moreover, more than enough space is left on the FPGA to implement this without any constraints. During this extension, the delay should be taken into account as the cancellation will come before the detection algorithm and therefore add delay to the overall detection time.

Another possible extension is the optimization of the current detection algorithm,

which is quite computational intensive. The speed of the algorithm can however be further increased by using more hardware or by rewriting the algorithm. Any optimization on the algorithm allows to increase the performance of the overall network, as collisions and interferers can be detected faster. Moreover, if the detection algorithm can be optimized towards sub-10 μs detection latencies, the implementation can also be used for faster PHYs like WiFi.

Detecting collisions at the receiver instead of at the transmitter is another possible extension. This would require to implement an algorithm that can detect if the current received bits are valid or not. Based on this information the transmitter can be notified. The notification channel can then be used for more than just collision information, other control information can be passed to the transmitter as well. This control information can include transmit power or modulation scheme changes for example.

Scaling the network even further and implementing the network layer as well, is an important possible extension of the work in this thesis. The network layer is something that can be run inside the Microblaze which is running on each of the USRPs. This would then allow for experiments with heterogeneous networks where not all devices are in-band full duplex capable. Moreover, this would also require some additions to the current MAC implementation like acknowledgments.

Instead of using the USRPs to scale the network, one could also use them in a MIMO configuration. MIMO combined with IBFD could give potentially a high gain in throughput. However, the complexity of the self-interference cancellation explodes as each MIMO antenna will create interference on all the other antennas. Canceling all this interference is non-trivial, however, with some clever precoding of the transmitted signal already part of the interference can be avoided. Nonetheless, it would be an interesting extension as the potential gains can be high.

The current prototype uses a rather slow IEEE 802.15.4 PHY, an extension could be to upgrade the physical layer to IEEE 802.11 or IEEE 802.11p. This would allow experiments not only with low power sensors but also with WiFi devices. Besides the changes to the PHY, changes to the collision detection algorithm are needed as a typical WiFi packet is much shorter then an IEEE 802.15.4 packet. Another PHY upgrade could be the move towards mmWave. This would require changes to the analog cancellation hardware as the EBD is currently limited to a maximum carrier frequency of 2.1 Ghz. Moreover, it could be worthwhile to rethink the cancellation architecture as antennas at mmWave are typically more directional. This could relax the cancellation requirements as less of the transmitted signal is leaked into the receiver chain.

Finally, let us discuss the necessary steps to put this new technology in future products. Three general steps need to be done to make the technology ready for production. First, the analog cancellation circuit needs to be integrated with the rest of the analog front-end. This step should be fairly straightforward as the current design is already state of the art. Second, the current FPGA implementation of the collision detection algorithm should be integrated with the hardware implementation of the PHY and MAC layer. This would require interfacing all the functional blocks but also designing the actual circuit. Finally, the optimization algorithm for the analog cancellation needs to be integrated either in a co-processor or on the main processor, keeping the latency requirements in mind. Again, this step should be fairly straightforward as the current implementation is a pure C program.

Part II

Included papers

Chapter 7

Performance Analysis of In-Band Full Duplex Collision Detection

Submitted to:
Tom Vermeulen, Fernando Rosas, Brecht Reynders, Marian Verhelst and Sofie Pollin, "Performance Analysis of In-Band Full Duplex Collision Detection" in *IEEE Transactions on Networking*, 2017

Abstract

With the massive growth of wireless devices comes an increase in number of collisions and interference. This has a negative effect on throughput and energy consumption. In this paper, we propose and analyze a novel in-band full duplex collision and interference detection scheme, comparing both throughput and energy efficiency with traditional half duplex and symmetric full duplex transmissions. The main strength of our analysis is that we consider a realistic protocol and overhead and measurement-based self-interference. Our results indicate that our proposed collision detection scheme can provide important gains in terms of throughput and energy efficiency when the number of nodes of the network grows. Moreover, our results show that our scheme allows up to 45% more nodes in the network for the same energy per bit compared to half duplex and symmetric full duplex suggesting that this could be an enabling technology to achieve efficient very dense wireless networks.

7.1 Introduction

Network densification has been identified as one of the major challenges for future communication systems, as the increasing number of wireless mobile devices generates unprecedented coexistence problems [77]. The ongoing densification in space and frequency forces the devices to compete for the increasingly scarce communication resources [78], raising the amount of interference and the frequency of transmission collisions. Moreover, interference and collisions are specially difficult to control because of the hidden terminal problem, which is a consequence of the limited sensing ability of wireless devices [19]. Collisions and interference, aggregated by the hidden terminal problem, are fundamental performance bottlenecks for dense wireless networks [79, 80], like wireless sensor networks.

Collisions, coming from nodes using the same MAC protocol, and interference, coming from nodes outside the network, waste valuable transmission time and radiated power, having a negative impact on the energy efficiency, throughput and delay of the system. Moreover, existing MAC protocols rapidly become inefficient in dense wireless networks. For example, the collision avoidance mechanism of the well known CSMA technique is only efficient for a small number of nodes, hence the densification of CSMA networks decreases the total network throughput and increases the delay [81]. In addition, it has been shown that the energy consumption of networks based on CSMA increases exponentially with respect to the number of nodes and the total network throughput [16]. The main problem is that —unlike in wired systems— wireless nodes under this

scheme are unable to detect collisions in real-time and abort their transmission. Moreover, the lack of instantaneous collision feedback makes them waste the scarce wireless medium with transmissions that are not going to be decoded anyway.

Several state-of-the-art techniques have been proposed to improve the performance of CSMA networks by detecting collisions early. In [26], a collision notification scheme is presented, where the receiving node detects collisions and transmits a unique signature to notify the transmitter, who is constantly looking for this signature. Although this scheme provides some throughput improvements by detecting collisions due to the shortened collision time, it does not solve the hidden terminal problem. In [25], a wireless CSMA/CD scheme is presented, the collisions are detected by randomly switching of the transmitter during the packet and listening for collisions. The authors show the improvements in throughput, however, this introduces a significant overhead because of the switching between transmit and receive. Another MAC protocol, presented in [24], uses pulses in an out-of-band control channel for collision detection and a clear to send signal to avoid hidden terminals. The protocol uses extra bandwidth and requires two transceivers on different frequencies. Moreover, the protocol is not able to detect interference from other networks if they don't follow the same protocol. In [82] another collision detection scheme for WiFi is analyzed, showing that it can increase the throughput. They conclude that collision detection can increase the throughput if the MAC parameters are chosen correctly. However, they don't analyze the performance for dense networks and only look at the throughput of their scheme, hence neglecting the full system energy consumption.

Ideally, collisions should be detected as soon as possible in order to abort the transmission, saving energy and bandwidth. Detecting collisions while transmitting poses a number of important technical challenges, as for this task the transmitted signal is experienced as self-interference, which may be many orders of magnitude larger than the colliding signal that arrives after being attenuated by path loss [35]. This problem is similar to the one found in in-band full duplex, where devices transmit and receive data at the same time and on the same frequency. Key ingredients for making IBFD work are efficient methods of SI cancellation, which usually combine different analog and digital techniques. If the SI signal is canceled below the noise floor, IBFD can potentially double the physical layer throughput without increasing the bit error rate and the frequency usage [11]. Recently, the practical feasibility of IBFD transceivers that efficiently cancel the SI has been demonstrated in a number of testbeds [37, 11].

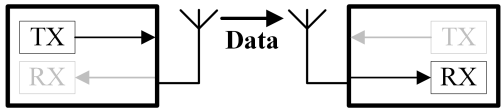
From the previous discussion, it is natural to ask if IBFD technology could be used to design a MAC protocol for dense wireless networks. Our main idea,

which is explored in this paper, is to use SI cancelation not to increase the bidirectional throughput, but to implement a mechanism of real-time collision detection and notification. In this way, the receiver can sense collisions during reception as soon as they take place, and provide real-time feedback to the transmitter over the full duplex link if the incoming frame has to be aborted. This provides two direct benefits: save energy at both colliding nodes and free up the wireless spectrum for other users, i.e. increase throughput.

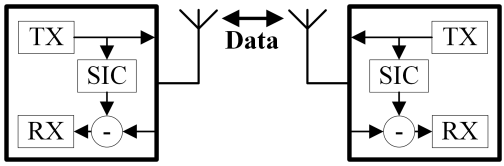
Our previous work [30, 83, 17] has shown some of the promising benefits that such a protocol might introduce for wireless sensor networks in order to avoid the collision and interference waste. A slotted version of the protocol was discussed in [30], where it was shown how an IBFD-based collision detection scheme can improve the energy efficiency and reduce the delay of networks in saturated and unsaturated conditions when compared to half duplex networks. This work was extended in [17], where an unslotted version of the protocol was presented and analyzed. Finally, [83] showed how the energy efficiency can be further improved by introducing a scalable SI cancellation module, which dynamically enables or disables different cancellation stages.

To complement our previous work, the main goal of this paper is to analyse not only the energy efficiency but also throughput of our IBFD collision detection scheme and compare it with traditional full duplex bidirectional transmissions. We also aim to compare these two schemes with half duplex CSMA transmissions, in order to generalize the ideas discussed in [30, 83, 17] within a single novel theoretical framework. Moreover, our previous work was based on numerical evaluations of [18], while the results of this framework have been verified with novel ns-3 simulations and measurement. This allows us to compare these schemes not only in saturated traffic but also in unsaturated traffic. Our results show that our IBFD collision detection scheme outperforms the other two schemes in terms of energy efficiency for high throughput networks and when the number of users in the network is high. Looking at throughput, we see that traditional full duplex transmissions are only beneficial in symmetric saturated traffic conditions. In all other scenarios, our proposed collision detection scheme has similar throughput.

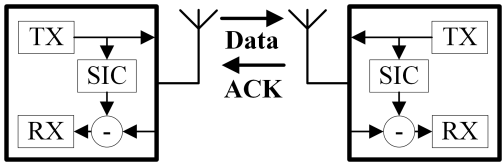
This paper is organized as follows, Section 7.2 presents an overview of the IBFD schemes considered in this paper. In Section 7.3 our performance model is presented, followed by some simulation results in Section 7.4. Next, in Section 7.5, we formulate a mathematical model that describes the energy consumption of the three schemes under diverse traffic conditions, and symmetric or asymmetric uplink versus downlink data flow, followed by an analysis of the energy efficiency in Section 7.6. Finally, in Section 7.7, we draw some conclusions.



(a) HD-CSMA/CA: Half duplex scheme with either the transmitter or the receiver active.



(b) FD-CSMA/CA: Full duplex scheme with data in both directions.



(c) FD-CSMA/CD: Full duplex scheme where data is send in one direction and collision information in the other.

Figure 7.1: Comparison of the three schemes considered in this paper.

7.2 Overview of half duplex and in-band full duplex schemes

In this paper, we do an energy and throughput comparison of half duplex transmissions with two different in-band full duplex schemes. Figure 7.1 introduces all three schemes considered in this paper. In the sequel, Section 7.2.1 provides an overview of the legacy CSMA/CA protocol, followed by the full duplex version of the same protocol in Section 7.2.2. Section 7.2.3 introduces our proposed FD-CSMA/CD scheme.

7.2.1 HD-CSMA/CA

In contention-based systems like WiFi [13] and ZigBee [12] the most commonly used medium access control scheme is carrier sense multiple access with collision avoidance (CSMA/CA), which in this paper is denoted as half duplex carrier sense multiple access with collision avoidance (HD-CSMA/CA) (Figure 7.1a).

A half duplex transceiver activates either its transmitter or receiver and only one node can be transmitting at the same time. This scheme uses carrier sensing to ensure that the wireless medium is unoccupied before transmitting a packet. If the wireless medium is occupied, the device performs a *random backoff* by adding a random delay to the transmission to avoid colliding with the transmission that is taking place. The backoff time increases exponentially with each backoff to further avoid collisions. A collision occurs when either the carrier sensing mechanism fails due to hidden terminals or when two or more nodes sense the channel unoccupied. In both cases, the sensing nodes start their transmission at the same time.

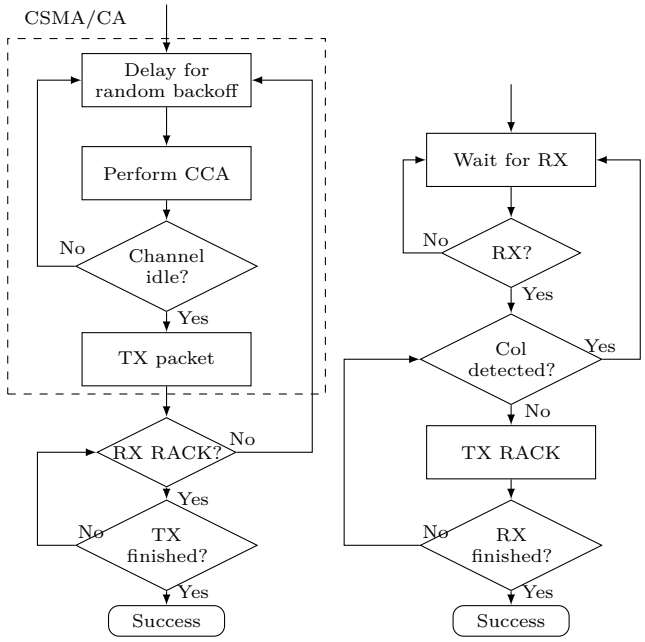
7.2.2 FD-CSMA/CA

CSMA/CA can also be implemented over full duplex transmissions, which is denoted as full duplex carrier sense multiple access with collision avoidance (FD-CSMA/CA) (Figure 7.1b). Here self-interference cancellation (SIC) is used to remove all or part of the self-interference from the received signal. This creates a self-interference free downlink channel, which can be used to transmit data in both directions [42]. Besides its transmitter chain, a full duplex transceiver needs to activate its receiver chain and self-interference cancellation. If both nodes in a link have data to send to each other the receiver will also become a transmitter, potentially doubling the link throughput. If only one of the participating nodes has data to send, this scheme reduces to the regular HD-CSMA/CA case.

Interestingly, this scheme is able to mitigate the hidden terminal problem when there is a full duplex transmission, as the signal radiated by the receiver creates a busy signal to its neighbouring nodes. Unfortunately, this only works when the receiver has data to send. Besides mitigating the hidden terminal problem, FD-CSMA/CA also improves access point (AP) fairness, as it gives the AP a downlink slot every time a node acquired the medium [42]. Unfortunately, it does not reduce the collision time and therefore still wastes resources during collisions and interference, because collisions are still fully transmitted.

7.2.3 FD-CSMA/CD

Our proposed scheme uses CSMA with collision detection over full duplex links, which is denoted as FD-CSMA/CD. The setup uses self-interference cancellation to clean up the received signal, as described in Section 7.2.2. In this setup the uplink is used to transmit sensor data to the central node, while the downlink can be used as an instantaneous control channel (Figure 7.1c). Therefore, data



(a) Transmitter side (b) Receiver side

Figure 7.2: Flowchart of the FD-CSMA/CD algorithm

is transfered following a half duplex protocol, and the full duplex capability is entirely devoted to detect collisions and provide instantaneous feedback information. Our earlier work [30, 17] proposed the FD-CSMA/CD scheme to leverage this information to optimize throughput, delay and energy.

Our FD-CSMA/CD protocol is described in Figure 7.2. At the transmitter side (Figure 7.2a), the node follows the standard CSMA/CA protocol to transmit the packet. At the receiver side (Figure 7.2b), the node starts its collision detection, upon receiving the beginning of the packet. If it doesn't detect any collisions it starts transmitting a real-time acknowledgment (RACK) on the downlink slot. This is then received by the transmitting node who keeps on transmitting as long as it receives the acknowledgment. In this paper, we assume it is possible for the receiving node to detect collisions during the reception of the packet, which can be done using physical layer information, like the confidence levels from SoftPHY [84] for example.

Our algorithm on a timescale is presented in Figure 7.3. When there is no collision (Figure 7.3a), the receiver immediately after decoding the header starts transmitting the RACK and the transmission continues. When a collision

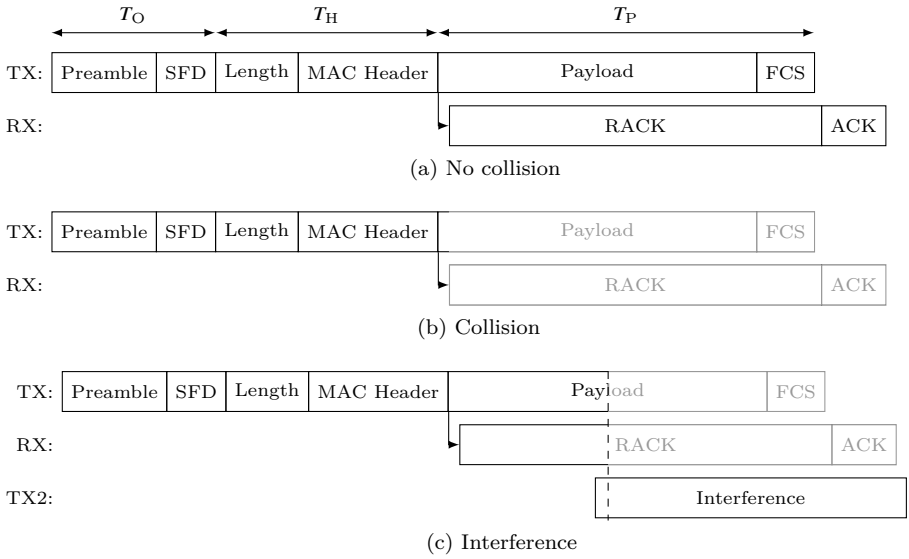


Figure 7.3: Timeframe of the FD-CSMA/CD algorithm

happens (Figure 7.3b), either the packet won't be detected or the receiver will detect both the packet and the collision. In both cases the receiver won't transmit the RACK and the transmitter will stop its transmission. When interference is present (Figure 7.3c), the receiver will detect this and stop its RACK transmission. The transmitter will react to this by aborting its transmission. The previous discussion clearly shows the difference with other schemes like CSMA/CN [26]. In this case of CSMA/CN, the notification is only transmitted when a collision is detected. However this poses two problem, first if there is a collision on the header, the packet won't be correctly detected and no notification will be transmitted. Second, if there is a collision on the notification, the transmitter won't be able to detect this. Our scheme does not have these problems as in both situations the transmitter will abort its transmission.

The instantaneous acknowledgment not only enables collision detection at the transmitter side but also mitigates the hidden terminal problem, because the receiver is continuously transmitting feedback information. These feedback packets are sensed by the surrounding nodes and makes them defer their transmission. If the network consists of a combination of IBFD and legacy, half duplex, nodes, then these half duplex nodes will also find the channel occupied by the instantaneous acknowledgment, therefore the hidden terminal problem is also solved for these nodes.

It is clear that if one can decrease the collision time, one increases the overall throughput of the system. This will be shown in the next section. However, to achieve this there is an added energy cost for the instantaneous acknowledgment compared to half duplex. These trade-offs will be investigated in Section 7.5.

7.3 Performance model

In this section, we develop a model for the assessment of the performance of the three types of wireless links studied in Section 7.2, taking into account realistic models for the self-interference. First we analyze the different components that influence the performance. Subsequently, we end this section with a throughput model.

In the following sections we focus on IEEE 802.15.4 nodes as this standard is often used in wireless sensor networks. To reduce the modeling complexity, we focus on unacknowledged packets, where packets are always detected. Moreover, the material presented here focuses on uncoded transmissions, which is consistent with the IEEE 802.15.4 standard, while an extension to coded transmissions is possible using the results from [85]. Finally, we consider networks in a star topology with one central node, leaving issues related to routing for future work.

In this section, we look at how decoding errors, collisions and interference affect the link performance, defined by the average time required per successfully transmitted bit. The average time per successfully transmitted bit ($\bar{T}$) reflects the time a wireless device is actively using the wireless spectrum to transmit data, and can be expressed as

$$\bar{T} = T_{\mathrm{b}}\bar{\tau}_{\mathrm{d}} + T_{\mathrm{i}}\bar{\rho}_{\mathrm{i}} + T_{\mathrm{c}}\bar{\rho}_{\mathrm{c}}, \tag{7.1}$$

where the first term is due to decoding errors, the second due to interference and the final one due to collisions. $\bar{\tau}_{\mathrm{d}} \geq 1$ is the average number of *transmission trials* to decode the frame error-free given that there are no collisions or interference other than self-interference, $\bar{\rho}_{\mathrm{i}} \geq 0$ is the average number of *retransmissions* due to interference given that there are no collisions and $\bar{\rho}_{\mathrm{c}} \geq 0$ is the average number of *retransmissions* due to collisions. T_{b} is the time per transmitted bit as defined by the physical layer, T_{i} is the time per bit lost before the interference is detected and T_{c} is the time per bit before a collision is detected.

Let us now look at the components that make up (7.1).

7.3.1 Transmission trials τ_d due to decoding errors

T_b is mainly dependent on parameters defined by the physical layer (PHY). T_b consists of the time to transmit the header, T_H, the payload, T_P and the overhead signals for channel estimation, synchronization, etc., T_O. Dividing the sum by the total amount of data bits in the payload, L_P, gives the airtime per payload bit

$$T_b = \frac{T_P + T_H + T_O}{L_P} \quad . \tag{7.2}$$

By denoting the physical layer symbol rate as R_s and the number of bits per symbol as b, one can rewrite T_b as

$$T_b = \frac{1}{bR_s}\left(1 + \frac{L_H}{L_P} + \frac{L_O}{L_P}\right) \quad , \tag{7.3}$$

where L_H is the number of bits of header and L_O correspond to the cost of the overhead —measured in bits.

$\bar{\tau}_d$ is also dependent on PHY parameters such as the channel statistics, SNR, modulation, code rate and frame size [86]. The SNR is not only dependent on the distance but also on the SI cancellation. We therefore model the remaining self-interference based on measurements from our IBFD prototype using an electrical balance duplexer [38] and a USRP RIO, as shown in Figure 7.4.

In an IBFD system, the received signal after analog cancellation, but before digital cancellation (cf. Figure 7.4), can be expressed as

$$y = hx + h_{SI}x_{SI} + w_n, \tag{7.4}$$

where hx is the useful signal coming from the other node, $h_{SI}x_{SI}$ is the remaining self-interference and w_n is the corresponding additive noise term. In general, the interference and noise on the useful signal x is dominated by the self-interference signal.

After digital cancellation, the received signal can be written as

$$y_d = hx + w'_n, \tag{7.5}$$

where we assume the interference and noise will no longer be dominated by the self-interference and $w'_n = w_n + (h_{SI} - \hat{h}_{SI})x_{SI}$. In this section we are interested in finding the properties of w'_n.

To measure the self-interference, the IBFD architecture from [71] is used, as shown in Figure 7.4. The setup uses the National Instruments USRP RIO platform [87] for the baseband processing. To cancel the self-interference signal

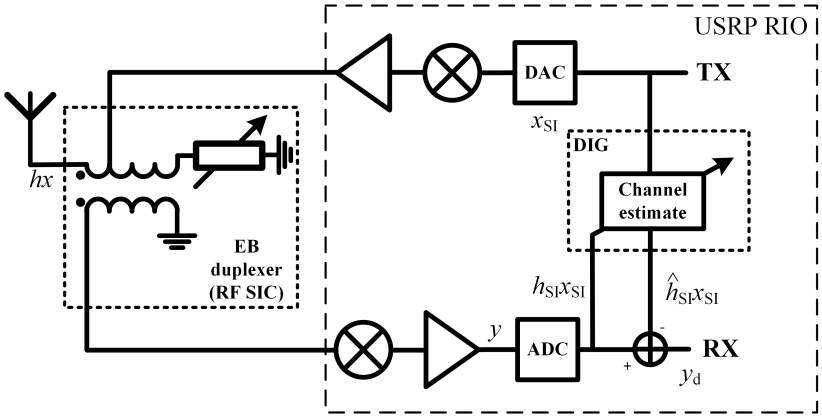
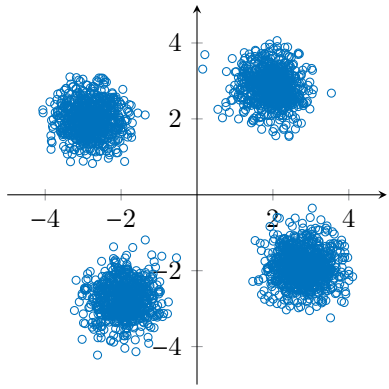


Figure 7.4: In-band full duplex architecture used in the measurements and mathematical modeling

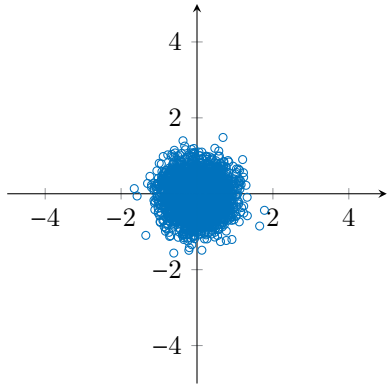
in the analog domain, a custom electrical balance duplexer (EBD) [38] is used. The EBD balances the impedance from the antenna in order to create an inverse copy of the SI signal. This inverse copy destructively interferes with the SI signal, achieving a cancellation of at least 50 dB at RF. Next, offline digital cancellation is applied, where a least squares estimate of h_{SI}, i.e. $\hat{h}_{SI}$, is obtained. Finally, the reconstructed signal $\hat{h}_{SI}x_{SI}$ is subtracted from the received signal. The combination of these techniques provides over 90 dB cancellation to keep the SNR on a similar level as in the half duplex case, which is confirmed by our measurements in Figures 7.5 and 7.6.

The USRP transmits a QPSK modulated signal with a sinc pulse shape. The bandwidth of the signal is 5MHz, consistent with the channel bandwidth of IEEE 802.15.4 [12]. The signal is transmitted through the EBD, which is configured to give a cancellation of around 50dB. The resulting signal is received again by the second front-end of the USRP. Both the transmitted and received signal as well as the noise w_n without any signals present, were logged. Next the signals are loaded into MATLAB, where a custom digital cancellation scheme is ran to estimate h_{SI}.

Figure 7.5 shows a comparison of the self-interference symbols before digital cancellation ($h_{SI}x_{SI} + w_n$) and after digital cancellation (w'_n). These figures show that the leakage from the QPSK transmit data is attenuated below the receiver noise floor, and hence prove that the assumptions made in (7.4) and (7.5) are valid. From the constellation diagram of Figure 7.5b it is not possible to distinguish the self-interference symbols which were still clearly visible before digital cancellation. However, it is not clear if the noise is dominated by



(a) Before digital cancellation ($h_{\mathrm{SI}}x_{\mathrm{SI}} + w_{\mathrm{n}}$)



(b) After digital cancellation (w'_{n})

Figure 7.5: Constellation graph of the measured signals

$(h_{\mathrm{SI}} - \hat{h}_{\mathrm{SI}})x_{\mathrm{SI}}$ or w_{n} and what the distribution of the noise is.

To determine this, the distributions of w'_{n} and w_{n} were compared. This can be done using the two-sample Kolmogorov-Smirnov test [88]. Applying this test on both signals shows that both distributions are very similar (asymptotic P-value of more than 0.8), i.e, w'_{n} is dominated by w_{n}. The histogram of the signal after digital cancellation (Figure 7.6) clearly follows a Gaussian distribution. Furthermore, most of the evidence found in the literature suggests that h_{SI} follows a Rician distribution*. We have validated this assumption using our digital cancellation scheme.

*For a thoughful discussion on this issue c.f. [89] and references therein.

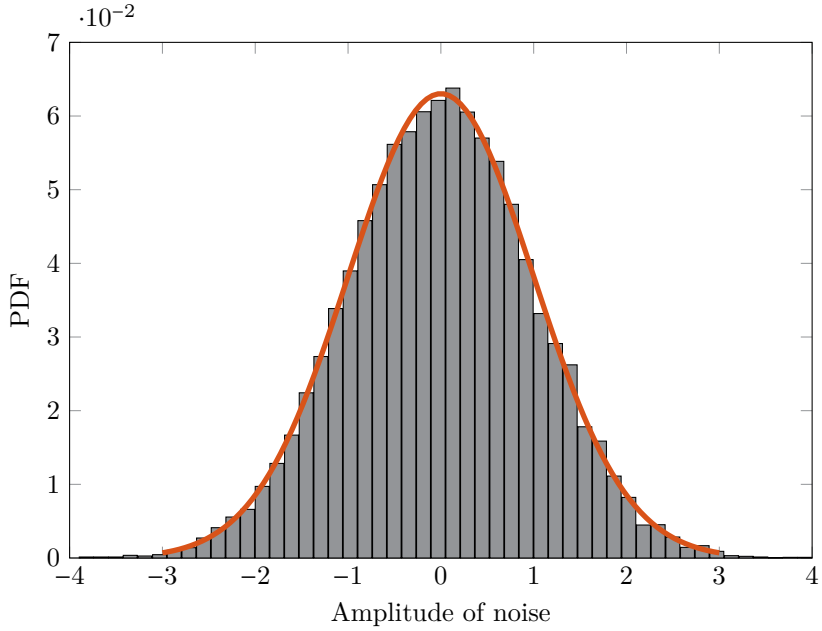


Figure 7.6: Histogram of the measured signal after digital cancellation (w'_{n}) with a gaussian PDF plotted on top

Following the measurement results, it is safe to assume that the noise w'_{n} is Gaussian distributed if analog and digital cancellation is applied. Based on the previous results, we model the signal-to-self-interference-and-noise ratio (SSINR or $\bar{\gamma}$) of a full duplex link as

$$\bar{\gamma} = \frac{P_{\mathrm{rx}}}{|h_{\mathrm{SI}}|^2 P_{\mathrm{tx}} + P_{\mathrm{n}}}, \tag{7.6}$$

for the case where only analog cancellation is active. Here P_{rx} is the received signal strength of the useful signal, P_{tx} is the transmitted signal strength of the self-interference signal and P_{n} is the noise power. The self-interference channel, h_{SI} can be considered constant over time during the packet transmission but it is dependent on the frequency. For the case where both analog and digital cancellation are active, we model it as,

$$\bar{\gamma} = \frac{P_{\mathrm{rx}}}{|h_{\mathrm{SI}} - \hat{h}_{\mathrm{SI}}|^2 P_{\mathrm{tx}} + P_{\mathrm{n}}} \approx \frac{P_{\mathrm{rx}}}{P_{\mathrm{n}}} \tag{7.7}$$

and therefore,

$$\bar{\tau}_{\mathrm{d}}^{\mathrm{HD/CA}} = \bar{\tau}_{\mathrm{d}}^{\mathrm{FD/CA}} = \bar{\tau}_{\mathrm{d}}^{\mathrm{FD/CD}}. \tag{7.8}$$

This section proves that even in the case of real, imperfect, self-interference cancellation it is safe to assume that the remaining self-interference after analog and digital cancellation is dominated by gaussian noise.

7.3.2 Retransmissions $\bar{\rho}_{\mathbf{i}}$ due to interference

The total time per payload bit required to detect interference coming from other networks, T_{i}, can be written also as $T_{\mathrm{i}} = \gamma_{\mathrm{i}} T_{\mathrm{b}}$, where $0 \leq \gamma_{\mathrm{i}} \leq 1$. In the case of HD-CSMA/CA and FD-CSMA/CA $\gamma_{\mathrm{i}} = 1$ as interference is only detected at the end of the packet. In the case of FD-CSMA/CD, as interference is usually uncorrelated with the ongoing transmissions, it can occur at any point during the frame, after which it is detected instantaneously. Hence all values $\gamma_{\mathrm{i}} \in (0, 1)$ are equally likely. For evaluation purposes one can consider an average value of $\gamma_{\mathrm{i}} = 1/2$.

It is to be noted that $\bar{\rho}_{\mathrm{i}}$ is independent of the chosen MAC scheme, as it is due to interferers that are uncorrelated with our own transmissions.

7.3.3 Retransmissions $\bar{\rho}_{\mathbf{c}}$ due to collisions

Following the above rationale, we can rewrite $T_{\mathrm{c}} = \gamma_{\mathrm{c}} T_{\mathrm{b}}$, where for the cases of HD-CSMA/CA and FD-CSMA/CA $\gamma_{\mathrm{c}} = 1$ as there is again no reduction in the time per bit. In the case of FD-CSMA/CD links, collisions can only occur in the beginning of the transmission. This is a consequence of the fact that there are no hidden terminals because of the instantaneous feedback information (c.f. Section 7.2.3). The detection time of a collision is therefore assumed to be equal to the decoding time of the header (assuming it is received with no interference from outside of the considered network). Hence, the time required to detect a packet collision per goodbit is

$$T_{\mathrm{c}} = \frac{L_{\mathrm{H}} + L_{\mathrm{O}}}{b R_{\mathrm{s}} L_{\mathrm{P}}} \quad . \tag{7.9}$$

Note that (7.9) can be rewritten using (7.3) as $T_{\mathrm{c}} = \gamma_{\mathrm{c}} T_{\mathrm{b}}$, with

$$\gamma_{\mathrm{c}} = \frac{L_{\mathrm{H}} + L_{\mathrm{O}}}{L_{\mathrm{P}} + L_{\mathrm{H}} + L_{\mathrm{O}}} \quad . \tag{7.10}$$

This parameter is the fraction of the time per bit, T_{b}, required to detect a collision. As in general L_{P} is much larger than L_{H} and L_{O}, (7.10) shows that $\gamma_{\mathrm{c}} \ll 1$.

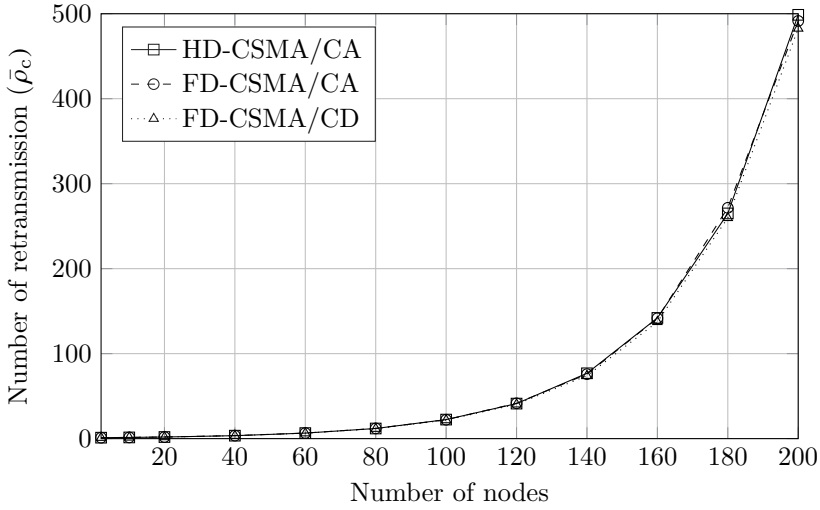


Figure 7.7: Validation of the number of retransmission due to collisions in saturated traffic conditions.

As this reduction in collision time reduces the congestion of the network, in general, $\bar{\rho}_c^{\mathrm{FD/CD}} \leq \bar{\rho}_c^{\mathrm{FD/CA}}$. Using ns-3, we validated this assumption for saturated traffic conditions (Figure 7.7).

7.3.4 Throughput model

By combining the information from the previous sections we find that the average time per successfully transmitted bit for HD-CSMA/CA is equal to

$$\bar{T}^{\mathrm{HD/CA}} = T_b(\bar{\tau}_d + \bar{\rho}_i + \bar{\rho}_c^{\mathrm{HD/CA}}). \tag{7.11}$$

For FD-CSMA/CA the average time can be expressed as

$$\bar{T}^{\mathrm{FD/CA}} = \frac{T_b}{2}(\bar{\tau}_d + \bar{\rho}_i + \bar{\rho}_c^{\mathrm{FD/CA}}), \tag{7.12}$$

if all transmissions occur in full duplex. And finally the average time per bit for FD-CSMA/CD is equal to

$$\bar{T}^{\mathrm{FD/CD}} = T_b(\bar{\tau}_d + \gamma_i\bar{\rho}_i + \gamma_c\bar{\rho}_c^{\mathrm{FD/CD}}). \tag{7.13}$$

By neglecting the time lost performing CCA and backoffs, we can define the throughput ($\mathcal{T}$) as the inverse of the average time per bit ($\bar{T}$), i.e.,

$$\mathcal{T} = (T_b \bar{\tau}_d + T_i \bar{\rho}_i + T_c \bar{\rho}_c)^{-1}, \tag{7.14}$$

therefore lowering the three components on the right hand side increases throughput.

From the previous formulas it is clear that full duplex transmissions and collision detection will always be better in terms of throughput as the average time per bit is lower, this will be validated in the next section. However, as we will see in Sections 7.5 and 7.6 there is a trade-off in terms of energy efficiency due to the increased power consumption of an IBFD transceiver.

7.4 Performance results

In this section, we perform ns-3 simulations of IEEE 802.15.4 nodes in a star topology and compute the average number of transmission trials due to collisions, $\bar{\rho}_c$. FD-CSMA/CA and FD-CSMA/CD was implemented on top of the existing IEEE 802.15.4 code of ns-3 version 3.22. In the physical layer, we added extra support for full duplex communication, while keeping the interface between the medium access layer and PHY layer identical. In the medium access layer, we added support for our proposed schemes. In any mode, the MAC layer asks the physical layer first to go to RX for assessing a clear channel (CCA). When CCA has successfully ended, the physical layer is asked to go to idle mode and switch TX on. Then, after the MAC header is sent or received, the full duplex nodes switch to full duplex mode. If everything is fine at the receiver side, the receiver answers with either a real-time acknowledgment or a packet for the sender for respectively FD-CSMA/CD or FD-CSMA/CA. If nothing is received, FD-CSMA/CD turns off the transceiver and starts over, while the FD-CSMA/CA scheme switches to half duplex mode.

The set-up of our simulation is as follows: traffic is generated in all nodes as specified below, and all nodes can hear each other perfectly as they are close enough to each other. The rest of the parameters are detailed in Table 7.1. The parameters are consistent with the IEEE 802.15.4 standard.

Each packet arrival is poison distributed. The three schemes are compared in four throughput scenarios

Table 7.1: Parameters used for ns-3 simulations

Parameter	Value
Frame Header — L_H	8 bytes[‖]
Payload length — L_P	90 bytes[‖]
Overhead — L_O	5 bytes[‖]
Bit per symbol — b	2[‖]
Symbol rate — R_s	125 kS/s[‖]

Source: [‖][12].

1. **Saturated, symmetric:** both access point and nodes receive a new packet every 5 ms[†];
2. **Saturated, asymmetric:** nodes receive a packet every 5 ms and the access point every 500 ms (i.e. every 100th packet, the access point sends a packet back);
3. **Unsaturated, symmetric:** both access point and nodes receive a new packet every 6 s;
4. **Unsaturated, asymmetric:** nodes receive a packet every 6 s and the access point every 60 s (i.e. every 10th packet the access point sends a packet back).

The throughput, calculated from the number of packets received in ns-3, from the first two scenarios is shown in Figure 7.8. Our simulations confirm that in symmetric saturated traffic conditions, FD-CSMA/CA can get double the throughput of HD-CSMA/CA. Figure 7.8 also shows that the shorter collision time means that the throughput of FD-CSMA/CD is higher than in the HD-CSMA/CA case. In asymmetric saturated traffic conditions, the performance of HD-CSMA/CA and FD-CSMA/CD does not change, however, FD-CSMA/CA loses most of its gain due to the fact that most of the transmissions no longer occur in full duplex.

An important benefit however of FD-CSMA/CA is AP fairness, shown in Figure 7.9. In this scheme, the AP always has a slot to transmit data to the nodes, and therefore is able to achieve true symmetric traffic conditions. This is in contrast to the other schemes where the AP has the same amount of transmit opportunities as the other nodes, i.e., $1/N$.

[†]Each transmission, including CCA and ACK is around 5 ms, therefore the nodes will always have at least one packet in their buffer.

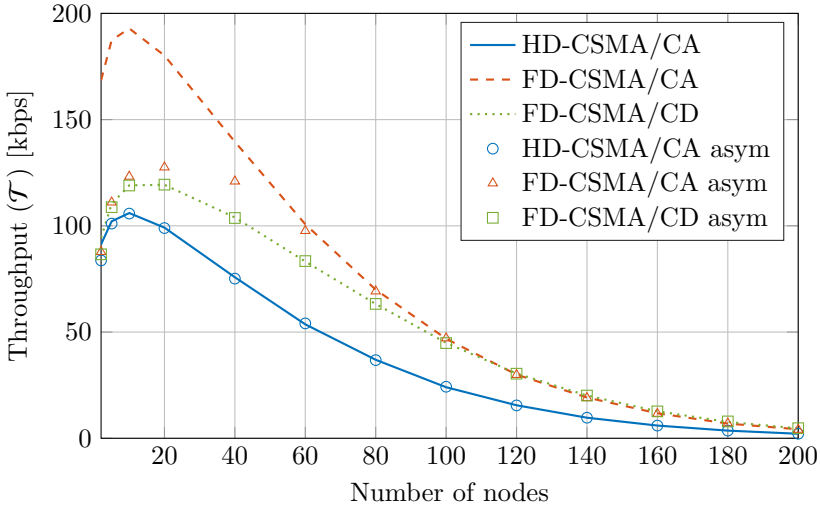


Figure 7.8: Saturated scenarios (1&2): FD-CSMA/CA increases the throughput in saturated traffic conditions by 2 compared to HD-CSMA/CA. The increase of FD-CSMA/CD compared to HD-CSMA/CA is between 1.2 and 2.

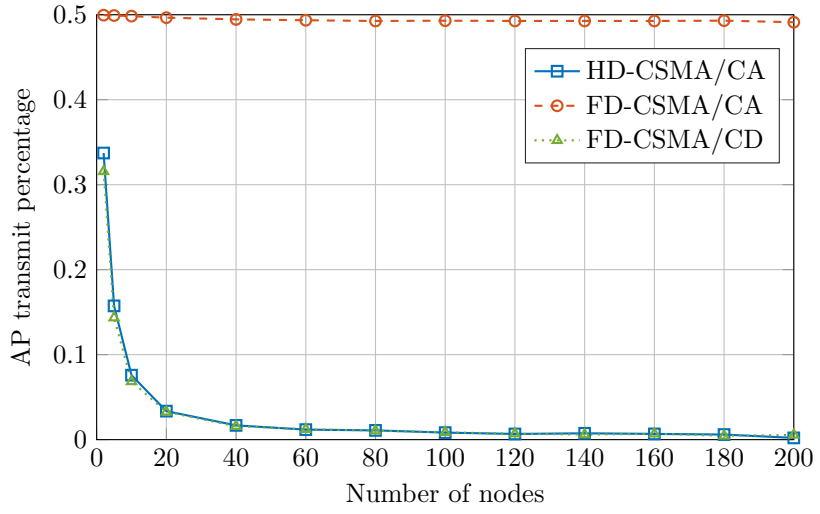


Figure 7.9: The AP fairness in symmetric traffic conditions shows that FD-CSMA/CA achieves a fair 0.5 downlink/uplink ratio in all situations while the other two schemes follow a 1/N curve.

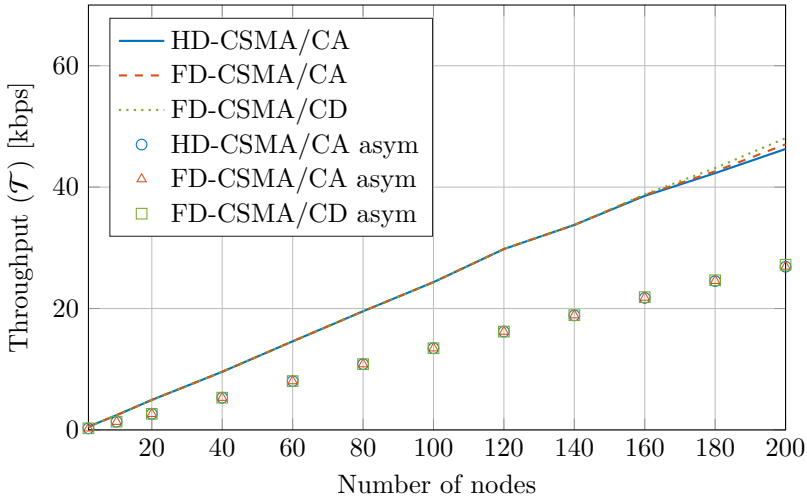


Figure 7.10: Unsaturated scenarios (3&4): The throughput in unsaturated traffic conditions increases linearly in all schemes. There is no difference in throughput between the different schemes.

Comparing the throughput for the different protocols in the unsaturated case (Figure 7.10), we see that for all three protocols the throughput linearly increases as more and more nodes enter the network. Next, the network starts to get saturated. We find that the throughput benefit of FD-CSMA/CA is not existent.

We can therefore conclude that due to the shorter collision time the throughput of FD-CSMA/CD is higher than HD-CSMA/CA in the saturated scenario and the difference with FD-CSMA/CA is in most cases small. Let us now look at how the three schemes compare in terms of energy consumption.

7.5 Energy model

In this section we develop a model for the energy consumption of the three types of wireless access studied in Section 7.2[‡]. The question we are trying to answer is if the increased energy consumption of an IBFD transceiver can be compensated for by the increased throughput. First, Section 7.5.1 presents the energy modeling of half duplex CSMA/CA transmissions, which is then extended

[‡]This model is an extension of what was presented in [83] and [17].

in Sections 7.5.2 and 7.5.3 for the cases of FD-CSMA/CA and FD-CSMA/CD links.

7.5.1 Performance of HD-CSMA/CA

Our goal is to estimate the energy per correctly transmitted data bit, which a node requires for exchanging data with the central node in a network with star topology. In the case of HD-CSMA/CA links, the nodes turn on their transmitter or receiver modules sequentially. In half duplex networks, the total transmit energy consumption per successfully transferred bit can be expressed as

$$\bar{\mathcal{E}}_{\mathrm{tx}}^{\mathrm{HD/CA}} = \frac{\left(P_{\mathrm{el,tx}} + P_{\mathrm{PA}}\right)}{\mathcal{T}} = \left(P_{\mathrm{el,tx}} + P_{\mathrm{PA}}\right) T_{\mathrm{b}} \bar{\tau}^{\mathrm{HD/CA}} \quad . \tag{7.15}$$

Above, the power consumption of the power amplifier is modeled as P_{PA}, the remaining power to transmit a packet as $P_{\mathrm{el,tx}}$ and T_{b} is the average air time per payload bit. The average number of transmission trials until a frame is decoded without errors, $\bar{\tau}^{\mathrm{HD/CA}}$, can be decomposed as

$$\bar{\tau}^{\mathrm{HD/CA}} = \bar{\tau}_{\mathrm{d}} + \bar{\rho}_{\mathrm{i}} + \bar{\rho}_{\mathrm{c}}^{\mathrm{HD/CA}} \quad . \tag{7.16}$$

The total energy required by a node to receive one bit of data successfully using HD-CSMA/CA transmissions can then easily be expressed as

$$\bar{\mathcal{E}}_{\mathrm{rx}}^{\mathrm{HD/CA}} = \frac{P_{\mathrm{el,rx}}}{\mathcal{T}} = P_{\mathrm{el,rx}} T_{\mathrm{b}} \bar{\tau}^{\mathrm{HD/CA}} \quad , \tag{7.17}$$

where $P_{\mathrm{el,rx}}$ is the electronic consumption of the receiver components.

To consider asymmetric traffic conditions, we introduce the parameter u that represents the percentage of bits transmitted in the uplink, and hence $1-u$ is the percentage of bits received in the downlink. Finally, the average consumption per information bit of a given node is given by

$$\bar{\mathcal{E}}_{\mathrm{b}}^{\mathrm{HD/CA}} = u\bar{\mathcal{E}}_{\mathrm{tx}}^{\mathrm{HD/CA}} + (1-u)\bar{\mathcal{E}}_{\mathrm{rx}}^{\mathrm{HD/CA}} \tag{7.18}$$

$$= \left[u(P_{\mathrm{el,tx}} + P_{\mathrm{PA}}) + (1-u)P_{\mathrm{el,rx}}\right] T_{\mathrm{b}} \bar{\tau}^{\mathrm{HD/CA}}. \tag{7.19}$$

7.5.2 Performance of FD-CSMA/CA

In contrast to HD-CSMA/CA, the nodes using FD-CSMA/CA keep both transmitter and receiver modules active. On top of this, the self-interference

cancellation (SIC) needs to be active as well. In the sequel, Section 7.5.2 analyzes the cost of the SIC module, and Section 7.5.2 summarizes our FD-CSMA/CA energy consumption model.

Energy consumption SIC

The SIC module is in general composed by an analog and a digital cancellation submodule, each of which have independent energy requirements. The optimal working point of both SIC components is dependent on impedance variations of the antenna and reflections from the environment. Therefore, we assume that these components need to be retuned every packet transmission. In our proposed architecture, the analog SIC module is composed of an EBD (c.f. Section 7.2). The EBD is a passive component that does not consume power during the frame transmission, only requiring energy during the tuning of the balance network. The energy consumption of the EBD per data bit per transmission trial is given by

$$\mathcal{E}_{\mathrm{EBD}} = \frac{P_{\mu\mathrm{C}} T_{\mathrm{EBD}}}{L_{\mathrm{p}}} \ , \tag{7.20}$$

where $P_{\mu\mathrm{C}}$ and T_{EBD} are respectively the power and time consumed by the microprocessor to find an optimal working point for the EBD.

The energy consumption of the digital SIC module per data bit is given by

$$\mathcal{E}_{\mathrm{DIG}} = P_{\mathrm{FIR}} \left(\frac{T_{\mathrm{FIR}}}{L_{\mathrm{p}}} + T_{\mathrm{b}} \right) \ , \tag{7.21}$$

where P_{FIR} is the power consumption of the finite impulse response (FIR) filter and T_{FIR} is the time it takes to estimate the channel. In contrast with the analog cancellation, digital cancellation consumes power not only while configuring the FIR filter but also during the frame transmission, this introduces an additional term (T_{b}).

Finally, the energy per bit for the full SIC scheme of our architecture is given by

$$\mathcal{E}_{\mathrm{SIC}} = \chi_1 \mathcal{E}_{\mathrm{EBD}} + \chi_2 \mathcal{E}_{\mathrm{DIG}} \coloneqq P_{\mathrm{SIC}}^{(\mathrm{d})} T_{\mathrm{b}} + \mathcal{E}_{\mathrm{SIC}}^{(\mathrm{s})} \ , \tag{7.22}$$

where χ_1 and χ_2 are indicator variables which are equal to 1 if the corresponding module is active and 0 if it's not. Above we are introducing the shorthand notation $P_{\mathrm{SIC}}^{(\mathrm{d})} = \chi_2 P_{\mathrm{FIR}}$ for the power consumption that corresponds to the costs that are proportional to the transmission time T_{b}, and $\mathcal{E}_{\mathrm{SIC}}^{(\mathrm{s})} = (\chi_1 P_{\mu\mathrm{C}} T_{\mathrm{EBD}} + \chi_2 P_{\mathrm{FIR}} T_{\mathrm{FIR}})/L_{\mathrm{p}}$ that is equal to the "static" energy consumption that does not grow with T_{b}.

Total energy consumption

In an IBFD link the receiver module of the transmitter is active during the transmission of data in the uplink to receive downlink data simultaneously. However, some of the electrical components of the transmitter front-end, like the clock generation, can be shared with the receiver front-end, therefore the power consumption of the electronic components is smaller than $P_{\mathrm{el,tx}} + P_{\mathrm{el,rx}}$. We introduce a parameter $0 < \alpha < 1$, such that power consumption of the electronic components of a full duplex transceiver is equal to,

$$P_{\mathrm{el,FD}} = P_{\mathrm{el,tx}} + \alpha P_{\mathrm{el,rx}} \ . \tag{7.23}$$

With this, and following a similar rationale than the one that led to (7.15), the energy consumption per transmitted goodbit of the FD-CSMA/CA can be modeled as

$$\bar{\mathcal{E}}_{\mathrm{tx}}^{\mathrm{FD/CA}} = \frac{P_{\mathrm{FD}}}{\mathcal{T}} + \mathcal{E}_{\mathrm{SIC}}^{(\mathrm{s})}\bar{\tau}^{\mathrm{FD/CA}} = \left[P_{\mathrm{FD}}T_{\mathrm{b}} + \mathcal{E}_{\mathrm{SIC}}^{(\mathrm{s})}\right]\bar{\tau}^{\mathrm{FD/CA}} \ , \tag{7.24}$$

where we are introducing the shorthand notation $P_{\mathrm{FD}} = P_{\mathrm{el,FD}} + P_{\mathrm{PA}} + P_{\mathrm{SIC}}^{(\mathrm{d})}$ and

$$\bar{\tau}^{\mathrm{FD/CA}} = \bar{\tau}_{\mathrm{d}} + \bar{\rho}_{\mathrm{i}} + \bar{\rho}_{\mathrm{c}}^{\mathrm{FD/CA}} \ . \tag{7.25}$$

Note that, as mentioned in Section 7.3, the SNR does not change between half and full duplex and therefore the terms $\bar{\tau}_{\mathrm{d}}$ and $\bar{\rho}_{\mathrm{i}}$ remain the same as in (7.16).

During a full duplex transmission, both transmitter and receiver modules are active, therefore the energy consumption to receive one goodbit is the same as to transmit it, i.e. $\bar{\mathcal{E}}_{\mathrm{rx}}^{\mathrm{FD/CA}} = \bar{\mathcal{E}}_{\mathrm{tx}}^{\mathrm{FD/CA}}$. When only one of the two nodes in the current transmission has data to send, this scheme reduces to the HD-CSMA/CA scheme. Therefore, by introducing u^{FD} as the percentage of full duplex transmission and using u as the percentage of half duplex transmission, the total average energy consumption per goodbit can be modeled as

$$\begin{aligned}\bar{\mathcal{E}}_{\mathrm{b}}^{\mathrm{FD/CA}} = [u^{\mathrm{FD}}\left(P_{\mathrm{FD}}T_{\mathrm{b}}/2 + \mathcal{E}_{\mathrm{SIC}}^{(\mathrm{s})}\right) + u(P_{\mathrm{el,tx}} + P_{\mathrm{PA}})T_{\mathrm{b}} \\ +(1 - u - u^{\mathrm{FD}})P_{\mathrm{el,rx}}T_{\mathrm{b}}]\bar{\tau}^{\mathrm{FD/CA}} \ .\end{aligned} \tag{7.26}$$

In the first term T_{b} is divided by 2 because twice the amount of bits can be transmitted in full duplex. When $u^{\mathrm{FD}} < 1$, the receiver or the transmitter modules are active and the scheme reduces to the half duplex case, hence the final two terms in (7.26). Note that $u^{\mathrm{FD}} \in [0, 1]$. It can be seen that highest energy efficiency of this scheme is achieved under equal bidirectional throughput, i.e. when the two nodes that compose the link continuously send data to each other, and hence $u^{\mathrm{FD}} = 1$.

7.5.3 Performance of FD-CSMA/CD

In the case of FD-CSMA/CD the electronic cost of transmitter, receiver and SIC modules are the same as for FD-CSMA/CA links. However, the energy cost of collisions is reduced as they are detected before the end of the transmission of the full frame, as discussed in Section 7.3.3. Therefore, the energy consumption per transmitted goodbit of the FD-CSMA/CD scheme can be expressed as

$$\bar{\mathcal{E}}_{\mathrm{tx}}^{\mathrm{FD/CD}} = \frac{P_{\mathrm{FD}}}{\mathcal{T}} + \mathcal{E}_{\mathrm{SIC}}^{(\mathrm{s})} \bar{\tau}^{\mathrm{FD/CD}} \tag{7.27}$$

$$= P_{\mathrm{FD}} T_{\mathrm{b}} \left(\bar{\tau}_{\mathrm{d}} + \gamma_{\mathrm{i}} \bar{\rho}_{\mathrm{i}} + \gamma_{\mathrm{c}} \bar{\rho}_{\mathrm{c}}^{\mathrm{FD/CD}} \right) + \mathcal{E}_{\mathrm{SIC}}^{(\mathrm{s})} \bar{\tau}^{\mathrm{FD/CD}} \ . \tag{7.28}$$

Above, $\bar{\rho}_{\mathrm{c}}^{\mathrm{FD/CD}}$ is the average number of retransmission due to collisions in the case of FD-CSMA/CD transmissions and, similarly to (7.16), $\bar{\tau}^{\mathrm{FD/CD}} = \bar{\tau}_{\mathrm{d}} + \bar{\rho}_{\mathrm{i}} + \bar{\rho}_{\mathrm{c}}^{\mathrm{FD/CD}}$.

Similarly as in the case of FD-CSMA/CA, the cost to transmit and receive data over a FD-CSMA/CD link is the same and hence $\bar{\mathcal{E}}_{\mathrm{rx}}^{\mathrm{FD/CD}} = \bar{\mathcal{E}}_{\mathrm{tx}}^{\mathrm{FD/CD}}$. However, FD-CSMA/CD links share data in a half duplex time division fashion, as in-band full duplex is purely used to receive real-time feedback information about collisions and interference while transmitting. Therefore, in contrast to FD-CSMA/CA, FD-CSMA/CD links cannot transmit and receive data at the same time. Accordingly, the total average energy consumption per bit shared over a FD-CSMA/CD link is

$$\bar{\mathcal{E}}_{\mathrm{b}}^{\mathrm{FD/CD}} = u \bar{\mathcal{E}}_{\mathrm{tx}}^{\mathrm{FD/CD}} + (1-u) \bar{\mathcal{E}}_{\mathrm{rx}}^{\mathrm{FD/CD}} \tag{7.29}$$

$$= P_{\mathrm{FD}} T_{\mathrm{b}} \hat{\tau}^{\mathrm{FD/CD}} + \mathcal{E}_{\mathrm{SIC}}^{(\mathrm{s})} \bar{\tau}^{\mathrm{FD/CD}} \ , \tag{7.30}$$

where we have introduced the shorthand notation $\hat{\tau}^{\mathrm{FD}} := \bar{\tau}_{\mathrm{d}} + \gamma_{\mathrm{i}} \bar{\rho}_{\mathrm{i}} + \gamma_{\mathrm{c}} \bar{\rho}_{\mathrm{c}}^{\mathrm{FD}}$ as the "reduced number of retransmissions". Comparing this with (7.19) and (7.26), (7.30) shows that the benefits provided by FD-CSMA/CD in terms of interference management can be suggestively represented as a reduction in the number of retransmissions required to achieve a correctly decoded frame. Finally, (7.30) also states that $\bar{\mathcal{E}}_{\mathrm{b}}^{\mathrm{FD/CD}}$ is independent of u, showing that the benefits introduced by FD-CSMA/CD are not affected by possible asymmetries in the traffic conditions.

Table 7.2: Parameters used for numerical evaluations

Parameter	Value
Tx electronic power — $P_{\text{el,tx}} + P_{\text{PA}}$	30.67 mW[*]
Rx electronic power — $P_{\text{el,rx}}$	35.28 mW[*]
Full duplex power ratio — α	0.7449[◇]
Microprocessor power— $P_{\mu\text{C}}$	13.53 mW[§]
FIR power — P_{FIR}	200 μW[‡]
EBD control time — T_{EBD}	128 μs
FIR control time — T_{FIR}	128 μs

From datasheet of [*] TI CC2420. Source: [◇] [30], [‡] [90], [§] [91].

7.6 Energy results

In this section, we present numerical evaluations that confirm the results presented in previous sections. For these results, we use the same simulations as in Section 7.4 and combine these with the model from the previous section. The extra parameters used for the numerical evaluations are detailed in Table 7.2. For T_{EBD}, we estimated this value from real-world experience with the electrical balance duplexer. The FIR power consumption corresponds to a 4-tap 10 bit filter in 90 nm technology [90].

First, the energy per bit of the three schemes is compared in saturated symmetric traffic conditions. This condition occurs for example during video conferencing where an equal amount of data is transmitted and received. Symmetric traffic gives the highest energy efficiency to FD-CSMA/CA, as all transmissions can take place in full duplex. However, our results show that FD-CSMA/CA is only more energy-efficient than FD-CSMA/CD when the number of nodes is low. In effect, Figure 6 shows that when the network density grows, the performance of FD-CSMA/CD scales more gracefuly, e.g., between 10 and 100 nodes, the energy per bit increases 18-fold for HD-CSMA/CA, while for FD-CSMA/CD this increase happens at 150 nodes. For the same energy consumption, FD-CSMA/CD allows up to 45% more nodes in the network compared to half duplex and full duplex transmissions. This is a consequence of the reduced cost of collisions provided by the instantaneous feedback. Comparing HD-CSMA/CA with FD-CSMA/CD, we see a very small difference in average energy per bit. This shows that the increased throughput of Figure 7.8 does not outweigh the increased energy consumption of a full duplex transceiver. When compared in asymmetric traffic conditions, the results do not vary significantly. The only difference is that u^{FD} in (7.26) becomes small and most transmission happen in half duplex for FD-CSMA/CA.

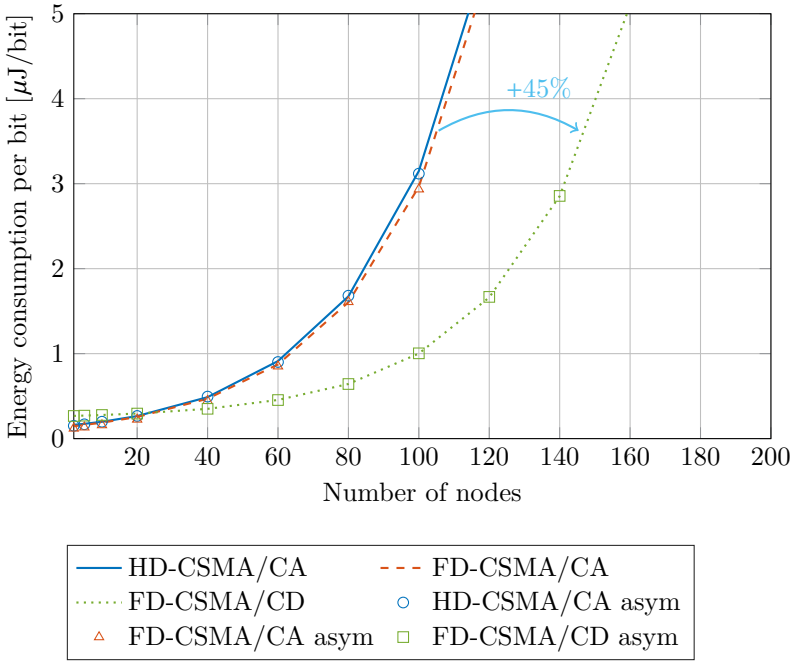


Figure 7.11: Saturated scenarios (1&2): The average energy consumption with no interference in saturated traffic conditions shows that FD-CSMA/CD is more energy efficient if more than 20 nodes are in the network compared to HD-CSMA/CA and FD-CSMA/CA.

Looking at the average energy per bit in unsaturated traffic conditions in Figure 7.12, it is clear that for HD-CSMA/CA and FD-CSMA/CA there is again only a small difference as u^{FD} is close to zero because there aren't any full duplex opportunities due to the empty packet buffers. Therefore the scheme is reduced to HD-CSMA/CA. The energy difference between two schemes and FD-CSMA/CD is small however for networks with less than 180 nodes and low traffic, it consumes more power.

7.7 Conclusion

In this paper we analyzed the performance and energy benefits of in-band full duplex collision and interference detection. To compare this scheme with half duplex transmissions and full duplex transmission, we implemented all three

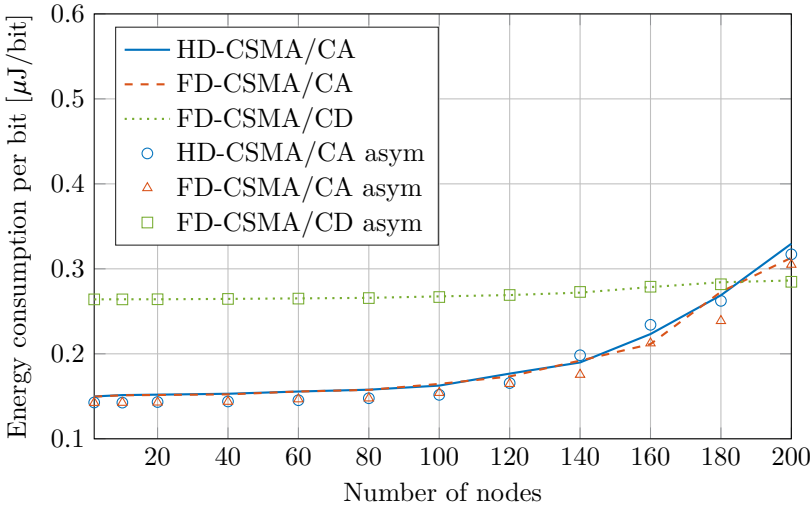


Figure 7.12: Unsaturated scenarios (3&4): In unsaturated traffic conditions, the average energy consumption with no interference shows only a minor difference between the three cases.

schemes in ns-3 and ran simulations to get the throughput and number of retransmissions. Next, we developed an energy model to compare the energy efficiency of the three schemes.

Our results suggest that in terms of throughput, under saturated traffic, FD-CSMA/CA performs best. However due to the shorter collision time the difference in throughput with FD-CSMA/CD disappears when more than 100 nodes are active in the network. Both schemes perform better than HD-CSMA/CA. In unsaturated traffic, the throughput difference between collision detection and full duplex transmissions becomes neglectable.

Looking at the energy efficiency, we can conclude that with saturated traffic, FD-CSMA/CD outperforms the other two protocols in terms of energy per bit. For the same energy per bit, it allows 45% more nodes than HD-CSMA/CA and FD-CSMA/CA in the network. FD-CSMA/CA performs better for a low number of nodes, achieving the lowest energy per bit of the three schemes. In unsaturated traffic, the difference between the three schemes is small.

To summarize, the proposed collision detection scheme is capable of reducing the energy consumption and increase the number of nodes in the network, while maintaining a high throughput that outperforms half duplex systems. Therefore, this technology might be an attractive alternative to answer the networking

challenges of future very dense wireless networks.

Chapter 8

Towards Instantaneous Collision and Interference Detection using In-Band Full Duplex

Published in:
Tom Vermeulen, Mihir Laghate, Ghaith Hattab, Danijela Cabric and Sofie Pollin, "Towards Instantaneous Collision and Interference Detection using In-Band Full Duplex" in *IEEE International Conference on Computer Communications*, 2017

Abstract

Wireless devices are ubiquitous nowadays and, since most of them use the same unlicensed frequency bands, the high number of packet losses due to interference and collisions degrade performance. Reliability, energy consumption, and latency are key challenges for future dense networks. Allowing the transmitter to take action, i.e., vacating the channel, as soon as a collision or interference is detected is crucial in improving these metrics. In-band full duplex radios enable the transmitter to simultaneously transmit packets and sense the spectrum for collisions and interference. This paper studies two important questions regarding transmitter-based collision and interference detection: (1) from an overall system perspective, does such detection outperform receiver-based detection and (2) which test statistic is the most accurate and sensitive at detecting collisions and interference. First, ns-3 simulations are used to show that transmitter-based detection reduces the energy consumption while improving the throughput in a typical star topology network. Next, we present a measurement-based study of four different techniques for transmitter-based collision and interference detection. In particular, we compare the energy detector with three goodness-of-fit tests in terms of probability of detection and false alarm. Our analysis shows that transmitter-based detection can detect between 80% to 100% of the collisions and interference occurring at the receiver, depending on the distance between the transmitter and the receiver. Of those detectable by the transmitter, our measurement results show that goodness-of-fit tests can detect nearly 100% of the collisions and have at least 10 dB better sensitivity as compared to the commonly proposed energy detection test. In general, the proposed techniques can detect interfering signals that are up to 25 dB below the remaining self-interference power.

8.1 Introduction

In the last decade, we have witnessed an explosion of wireless technologies and devices, and we are increasingly connecting everything around us. Both Gartner [92] and Cisco [93] estimate that there will be over 20 billion of these connected devices by 2020. On top of this, new applications like the tactile internet [6] require extremely low delay (1 ms) and high reliability ($> 99.99\%$). Unfortunately, this explosive increase in number of devices leads to denser heterogeneous networks, which in turn leads to more interference and collisions [21, 16], as these devices primarily communicate using contention-based protocols over the unlicensed and almost fully occupied Industrial, Scientific, and Medical (ISM) spectrum.

Collisions occur in contention-based networks, while interference typically comes from other networks or devices that are in range and use the same frequency. The difference between them is that collisions most often occur at the beginning of the packet, while interference can occur anytime during the packet. As a result, collisions waste more energy and transmission time that could have been avoided by a timely discovery, hence it is more important to detect them instantly. In both cases, two or more wireless nodes transmit a signal that overlaps in time, at least partially. The receiving node will receive a superposition of the two signals and in most cases the packets will be dropped because they cannot be decoded simultaneously, which leads to a retransmission.

Collisions and interference have become the bottleneck both in dense and in heterogeneous networks since they severely affect the delay, reliability, and energy consumption. Without proper management of interference and collisions, wireless nodes are forced to retransmit the same data more frequently, which wastes available resources. In dense networks, these retransmissions increase exponentially with an increasing number of nodes [80]. This problem is further compounded in heterogeneous networks by the fact that different wireless standards are unable to sense each other [15]. In addition, these retransmissions drastically increase the energy consumption [16], and since the majority of these devices are battery-powered (e.g., smartphones and laptops), their lifetime is severely affected. Even more importantly, retransmissions significantly increase delay or decrease reliability for a given delay constraint [94]. Therefore, managing interference and collisions, as quickly as possible, has become a critical issue that needs to be addressed. This paper investigates methods for detecting collisions and interferers at the transmitter and, thus, making it possible to instantaneously vacate the channel if required.

Most literature focuses on receiver-based detection [26], as that is where collisions and interference occur. The fundamental problem with receiver-based detection is that a feedback channel is necessary to notify the transmitter, making it impossible to take instantaneous action. The use of instantaneous feedback has been considered in [26], where an in-band collision notification is transmitted to the transmitter. It is shown that notifications can be correctly decoded if the difference in power between the signal of the transmitter and the notification is less than 36 dB. However, if the difference between the collision and the notification is larger than 12 dB, the notification will get lost. In addition, the scheme requires changes to both the receiver and the transmitter as the receiver needs to detect the collision in real-time.

Collision detection at the transmitter allows for instantaneous detection saving time and energy. Equally important, this approach does not require any cooperation from the receiver. Our system-level analysis shows that transmitters can detect between 80% to 100% of all collisions if the distance between the

nodes is less than 20 m. Detecting an interferer while transmitting is difficult, as the self-transmitted signal completely saturates the in-band receiver [35]. An emerging solution to tackle this problem is in-band full duplex (IBFD) [11], which enables two wireless nodes to simultaneously transmit and receive data on the same frequency by canceling out the self-transmitted (self-interference) signal. This paper uses IBFD for instantaneous collision detection and explores the fundamental decision bounds for various signals relying on measurements and various statistical detection methods. In particular, we investigate two different collision and interference detection methods: the energy detector (ED), and the goodness-of-fit methods. In the latter, we study three different tests: the Kolmogorov-Smirnov (KS) test [88], the Kuiper (KP) test [74], and the Anderson-Darling (AD) test [95]. These methods do not require any prior information of the interfering signal, rendering them practical and useful in environments where multiple networks of different technologies coexist.

Our measurement results, using an IBFD prototype and two USRPs, show that it is possible to reliably detect collisions and interference which are 25 dB below the self-interference after compensation by the IBFD system. More importantly, this can be done quickly, achieving an instantaneous detection within 250 μs, which is around 6% of the maximum packet length in IEEE 802.15.4 [12].

The rest of this paper is organized as follows. In Section 8.2, we provide some background on in-band full duplex. The system model and important design trade-offs are discussed in Section 8.3 followed by a system-level performance analysis in Section 8.4. Section 8.5 reviews the detection methods used in this work. Simulation and experimental results are presented in Section 8.6. Finally, conclusions are drawn in Section 8.7.

8.2 In-Band Full Duplex

Simultaneous transmitting and receiving or sensing has long been deemed impossible due to the large power difference between the transmitted and received signal. The self-transmitted signal, called the self-interference (SI), can be up to 100 dB more powerful than the signal being received simultaneously [35]. Thus, the strong self-interference completely saturates the receiver.

Knowledge of the transmitted signal can be exploited to help cancel the SI. In reality, however, we only know the digital representation of the SI signal before it goes through the transmitter chain. The distortions introduced by the transmitter and the large power, make canceling the SI challenging. First, a fraction of the signal needs to be canceled in the analog domain, to allow for enough dynamic range on the analog-to-digital converter to sample both the SI

and the signal we want to receive. Next, the remainder of the distortions will have to be estimated in the digital domain and then canceled.

Several works [11, 35, 37, 42] have demonstrated the feasibility of self-interference cancellation for in-band full duplex. In the analog domain, the cancellation can be done using a noise canceler [35] which changes the amplitude and phase of the transmitted signal and subtracts it from the received signal. A similar technique uses a balun [42] to create the canceling signal. This signal can also be created using an extra transmitter chain [37]. Other ways of canceling the self-interference in the analog domain use an electrical balance duplexer [96] or a vector modulator [39]. Analog cancellation typically allows cancellation between 50 and 70 dB of the self-interference. To cancel the SI even further, the remaining SI is removed in the digital domain. This is done by estimating the linear and non-linear components, reconstructing the remaining SI and subtracting it [11]. Digital cancellation is able to cancel up to 30 dB of the remaining self-interference [11, 37]. In this work, we only consider implementing the SI cancellation in the analog domain and rely solely on signal statistics and statistical detection theory to implement the collision detection in the digital domain.

8.3 System model

The system considered consists of three nodes: A transmitter-receiver pair, and an interferer as shown in Figure 8.1. The transmitter is equipped with an in-band full duplex architecture and transmits a signal to the receiver. The interfering node can belong to the same network as the transmitter, and thus the collision occurs due to mis-synchronization or contention. It can also belong to a different technology, and thus interference happens due to the lack of coordination between these different networks. Therefore, the transmitter node must transmit to the receiver and simultaneously sense the environment to determine the presence or absence of other nodes in order to detect collisions and interference. This concurrent transmission and sensing is possible via the IBFD architecture implemented at the transmitter. We assume only analog self-interference cancellation (SIC) is used and explore to what extent the system can work without digital self-interference cancellation and investigate how much digital processing gain our algorithms provide. Such an analog SIC can easily achieve 50 to 70 dB of cancellation [96, 11].

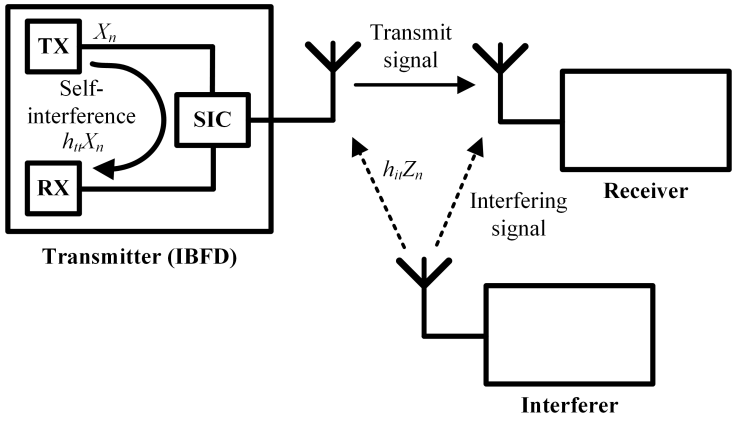


Figure 8.1: Overview of the system model and equivalent channel.

8.3.1 Mathematical Formulation of IBFD Collision and Interference Detection

The transmitter must determine the presence/absence of any interferer transmitting in the vicinity. Thus, we formulate the collision and interference detection problem as a binary hypothesis testing [97], where the transmitter decides which of the following hypotheses is true:

$$Y_n = \begin{cases} h_{tt}X_n + W_n & \text{if } \mathcal{H}_0 \\ h_{tt}X_n + h_{it}Z_n + W_n & \text{if } \mathcal{H}_1, \end{cases} \tag{8.1}$$

where n is the discrete time index, Y_n is the received signal at the transmitter node, X_n is the self-interference signal, Z_n is the interferer's transmitted signal, and W_n is an additive noise. h_{tt} is the self-interfering channel gain whereas h_{it} is the channel gain between the transmitter and the interference as illustrated in Figure 8.1. Both are assumed to be constant during the packet transmission. $\mathcal{H}_0$ and $\mathcal{H}_1$ denote the absence and presence of the interferer, respectively.

The transmitter node collects a block of N samples, and then processes them using a specific collision or interference detection technique in order to infer which hypothesis is true. If $\mathcal{H}_1$ is detected, the transmitter node immediately stops transmission; otherwise, the transmitter node continues transmitting to the receiver.

8.3.2 Sensitivity and Design Trade-offs

In this section, we formulate the detection and false alarm statistics of transmitter-based detection in terms of the sensitivity required.

Notation

From the mathematical formulation described in Section 8.3.1, we note that the transmitter transmits at power $\mathbb{E}[|X_n|^2]$ and receives SI of power $\mathbb{E}[|h_{tt}|^2|X_n|^2]$. We define $\delta = \mathbb{E}[|h_{tt}|^2]^{-1}$ as the SI cancellation. The interferer transmits at power $\mathbb{E}[|Z_n|^2]$, causes interference at the transmitter of power $\mathbb{E}[|h_{it}|^2|Z_n|^2]$, and at the receiver of power $\mathbb{E}[|h_{ir}|^2|Z_n|^2]$. For simplicity, we assume that the transmitter and receiver have the same noise power $\mathbb{E}[|W_n|^2]$.

Let the distance between the transmitter and receiver be d_{tr}, the distance between the receiver and the interferer be d_{ir}, and the distance between the transmitter and the interferer be d_{it}. We note that the propagation losses $\mathbb{E}[|h_{tr}|^2]$, $\mathbb{E}[|h_{it}|^2]$, and $\mathbb{E}[|h_{ir}|^2]$ can be expressed as functions of the distances d_{tr}, d_{it}, and d_{ir}, respectively, using the appropriate channel models such as the free space path loss model and ITU channel models. We denote the propagation loss due to such a channel model at distance d by $L(d)$ and its inverse map by $L^{-1}(\cdot)$.

Assumptions

Consider a system where the transmitter controls its transmit power such that the receiver experiences a signal-to-noise ratio (SNR) of γ, i.e.,

$$\frac{\mathbb{E}[|h_{tr}|^2]\mathbb{E}[|X_n|^2]}{\mathbb{E}[|W_n|^2]} = \frac{L(d_{tr})\mathbb{E}[|X_n|^2]}{\mathbb{E}[|W_n|^2]} = \gamma. \tag{8.2}$$

Assume that the receiver detects a collision or experiences interference if the received interference has power equal to or greater than the noise power, i.e., if

$$\mathbb{E}[|h_{ir}|^2]\mathbb{E}[|Z_n|^2] = L(d_{ir})\mathbb{E}[|Z_n|^2] \geq \mathbb{E}[|W_n|^2]. \tag{8.3}$$

Therefore, a radio causes interference at the receiver if it is located within a circle C_{ir} of radius

$$d_{ir} = L^{-1}\left(\frac{\mathbb{E}[|W_n|^2]}{\mathbb{E}[|Z_n|^2]}\right) \tag{8.4}$$

around the receiver.

Finally, we assume it is not possible to detect interferers below the noise floor and that the detection algorithm's processing gain β enables the transmitter to detect the interferer if the received power from the interferer is higher than a fraction β of the self-interference power, i.e., if

$$\begin{aligned}\mathbb{E}[|h_{it}|^2]\mathbb{E}[|Z_n|^2] &= L(d_{it})\mathbb{E}[|Z_n|^2] \\ &\geq \max\left\{\frac{\mathbb{E}[|X_n|^2]}{\beta\delta}, \mathbb{E}[|W_n|^2]\right\}. \end{aligned} \tag{8.5}$$

β depends on the algorithm used to detect the collisions and its parameters such as the number of samples used. In particular, the number of samples determines the time required to make a decision. The detection sensitivity can be defined as

$$\epsilon \triangleq \frac{1}{\beta\delta}, \tag{8.6}$$

i.e., the sensitivity improves when the SIC is improved or the processing gain of the detection algorithm is improved. Therefore, an interfering radio will be detected by the transmitter if it is located within a circle C_{it} of radius

$$d_{it} = L^{-1}\left(\frac{\max\left\{\epsilon\mathbb{E}[|X_n|^2], \mathbb{E}[|W_n|^2]\right\}}{\mathbb{E}[|Z_n|^2]}\right) \tag{8.7}$$

centered at the transmitter.

System-Level Performance Metrics

From Figure 8.2, we define the system-level collision detection probability $\mathbb{P}_{\mathrm{SD}}(d_{tr}, \epsilon)$ as the probability that an interferer is detected by the transmitter given that it causes harmful interference at the receiver, i.e., the probability that an interferer is located in $C_{it} \cap C_{ir}$ given that it is located within C_{ir}. A system-level false alarm probability $\mathbb{P}_{\mathrm{SF}}(d_{tr}, \epsilon)$ is defined as the probability that an interferer is detected by the transmitter given that it does not cause harmful interference at the receiver, i.e., an interferer is located in $C_{it} \backslash C_{ir}$ given that it is located within C_{it}. By assuming a uniform spatial distribution for the locations of the interferers within the appropriate circles, we can compute these probabilities as the ratios of the respective areas

$$\mathbb{P}_{\mathrm{SD}}(d_{tr}, \epsilon) = \frac{\mathrm{Area}(C_{it} \cap C_{ir})}{\mathrm{Area}(C_{ir})}, \tag{8.8}$$

$$\mathbb{P}_{\mathrm{SF}}(d_{tr}, \epsilon) = \frac{\mathrm{Area}(C_{it} \backslash C_{ir})}{\mathrm{Area}(C_{it})}. \tag{8.9}$$

The areas themselves can be computed using geometry.

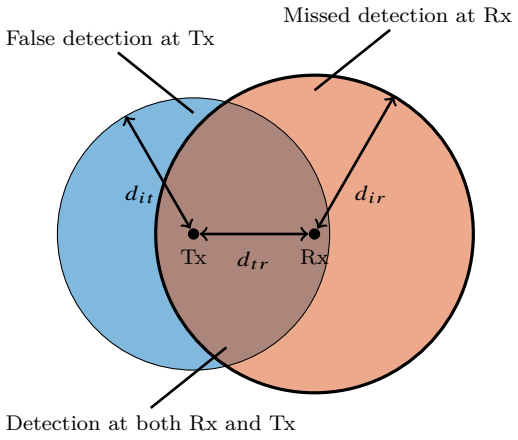
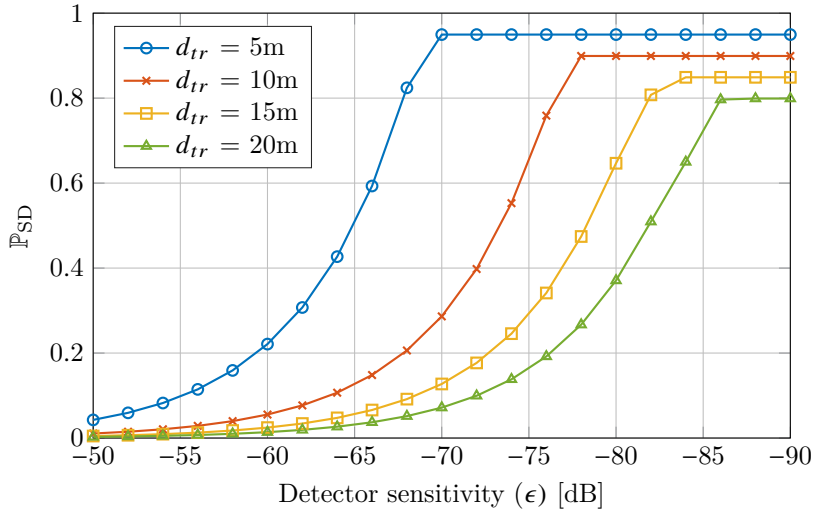


Figure 8.2: Interferers within the intersection of the two circles are correctly detected, others are either falsely detected or not at all.

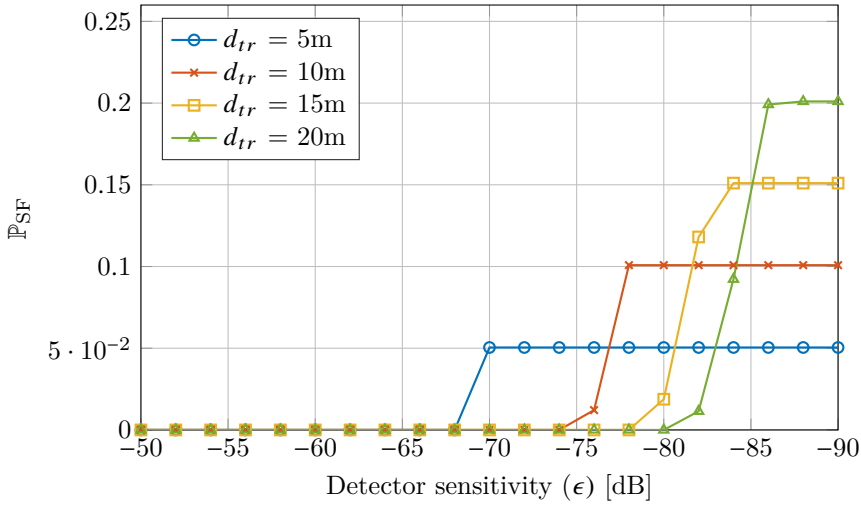
Analysis

A full duplex radio can control the SI cancellation δ and the processing gain β. Increasing either or both of these is considered to be a hard problem, as SI cancellation costs hardware and processing gain costs sensing time. Furthermore, (8.5) and (8.7) show that d_{it} is determined by the detector sensitivity, $\epsilon \triangleq (\beta\delta)^{-1}$, rather than the individual quantities β and δ. Therefore, we now determine the maximum possible value of ϵ that achieves the best performance in terms of $\mathbb{P}_{\text{SD}}$ and $\mathbb{P}_{\text{SF}}$.

Note the effect of the max operator in (8.7). For a given d_{tr}, Figure 8.3 shows that the detection performance saturates with decreasing ϵ. This comes from the fact that the performance is upper bounded by the noise power $\mathbb{E}[|W_n|^2]$ rather than the detector sensitivity ϵ, as we assumed no detection is possible below the noise floor. As long as the combination of the analog SIC δ and the digital processing gain β is below the noise floor, there is no point in improving them further. For a fixed ϵ, the same effect can be seen in Figure 8.4, where if d_{tr} is less than a certain d_{cutoff}, the performance flattens out. Hence, if $d_{tr} < d_{\text{cutoff}}$, $d_{it} = d_{ir}$ and $\mathbb{P}_{\text{SD}}$ and $\mathbb{P}_{\text{SF}}$ are independent of the value of ϵ as can be seen from Figure 8.4. This means that improving ϵ or increasing either β or δ does not increase performance. The value of d_{cutoff} depends on ϵ, the



(a) System level detection probability



(b) System level false alarm probability

Figure 8.3: Detection statistics versus detector sensitivity as a function of distance for the ITU channel model of an indoor residential environment at 2.4 GHz

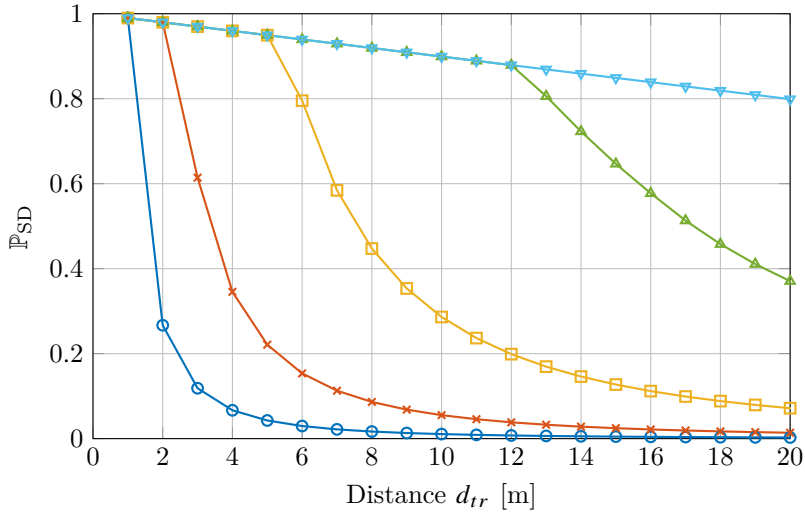


Figure 8.4: System level detection probability versus distance as a function of $\epsilon = -50$(—o—), -60(—×—), -70(—□—), -80(—△—), -90(—▽—) dB for the ITU channel model of an indoor residential environment at 2.4 GHz

noise and the signal power. It can be derived to be

$$d_{\text{cutoff}} = L^{-1}(\gamma\epsilon) = L^{-1}\left(\frac{\gamma}{\delta\beta}\right) \tag{8.10}$$

by using (8.2) and noting that

$$\epsilon\mathbb{E}[|X_n|^2 \leq \mathbb{E}[|W_n|^2]. \tag{8.11}$$

Finally, we draw some conclusions from these numerical results. Typically, IBFD radios can achieve a self-interference cancellation δ between 50 dB and 70 dB [11]. Let us assume $\mathbb{E}[|X_n|^2 = 0$ dBm and $\delta = 60$ dB, if the detection algorithm can achieve a processing gain $\beta = 25$ dB, then collision and interference detection at the transmitter has a sensitivity of $\epsilon = -85$ dBm and will be able to detect collisions with high probability even if the transmitter and receiver are located 15 m away from each other in an indoor environment.

8.4 System-level performance analysis

In this section, we discuss the advantages of transmitter-based detection to the network in terms of throughput and energy consumption. In existing literature,

several research papers investigated the higher layer benefits of collision detection. It is worth noting that none of these papers proposed practical solutions on implementing such a scheme.

8.4.1 Existing Work on Collision Detection based MAC protocols

Recent work has shown the system-level benefits of using IBFD for collision and interference detection [30, 17, 98]. The presented CSMA/CD protocol from these papers is very similar to the existing protocol. Wireless nodes will acquire the medium using carrier sensing. If the medium is acquired, the nodes transmit their packet. When a collision is detected, the transmitter aborts its transmission and enters the random backoff stage before it tries to acquire the medium.

In [30], a time-slotted CSMA-based IBFD collision reporting scheme is presented and analyzed. The scheme uses collision detection at the receiver and transmits collision feedback back over the full duplex link. The authors run simulations to show that their scheme performs better in terms of energy per bit when a certain number of nodes are active in the network compared to CSMA/CA networks. They also show that the average delay is lower using collision detection. The work is extended in [17] with a more detailed energy model for IBFD. The paper shows that the performance gain of instantaneous feedback increases with the number of nodes. However, both papers lack a practical analysis of the feasibility of detecting the actual collisions. Furthermore, this scheme requires feedback and therefore all nodes to be IBFD-capable.

An energy detection based collision detection scheme is analyzed in [98]. For a certain false alarm probability, the paper shows that collision detection can greatly improve the throughput compared to collision avoidance, however, the paper does not consider interference, moreover, perfect detection is assumed.

These papers show that wireless networks can benefit greatly from collision and interference detection, not only in terms of throughput, but also in terms of energy consumption and delay. However, they also show that there is insufficient results on the practical feasibility of such schemes.

8.4.2 Protocol Analysis

To analyze the system-level performance of our transmitter-based collision detection scheme, we implemented the CSMA-based collision detection MAC

protocol from [30] in ns-3 [20] and calculated the collision probability ($\mathbb{P}_{\text{col}}$). The ns-3 simulations simulate IEEE 802.15.4 sensornodes in a star topology in a worst case scenario where each node always has an uplink packet to send. In such a system, full duplex transmissions would not be beneficial as there are no packets in the downlink and therefore no full duplex opportunities. Using the system-level detection and false alarm probabilities from the previous section and the collision probabilities from ns-3, we can define four different states of the transmission:

- **Success ($\mathbb{P}_{\text{S}}$):** No collision and no false alarm;
- **Long collision ($\mathbb{P}_{\text{LC}}$):** Collision but not detected;
- **Short collision ($\mathbb{P}_{\text{SC}}$):** Collision and detected;
- **False short collision ($\mathbb{P}_{\text{FSC}}$):** No collision but false alarm.

Half duplex systems only have the first two states while systems with collision detection have all four states. The collision detection state probabilities are given by

$$\mathbb{P}_{\text{S}} = (1 - \mathbb{P}_{\text{col}})(1 - \mathbb{P}_{\text{SF}}) \tag{8.12}$$

$$\mathbb{P}_{\text{LC}} = \mathbb{P}_{\text{col}}(1 - \mathbb{P}_{\text{SD}}) \tag{8.13}$$

$$\mathbb{P}_{\text{SC}} = \mathbb{P}_{\text{col}}\mathbb{P}_{\text{SD}} \tag{8.14}$$

$$\mathbb{P}_{\text{FSC}} = (1 - \mathbb{P}_{\text{col}})\mathbb{P}_{\text{SF}}. \tag{8.15}$$

For half duplex these are equal to $\mathbb{P}_{\text{S,HD}} = 1 - \mathbb{P}_{\text{col}}$ and $\mathbb{P}_{\text{LC,HD}} = \mathbb{P}_{\text{col}}$. Following a similar analysis as in [18], we can now define the throughput as

$$S = \frac{\bar{B}[\text{average goodput per slot}]}{\bar{L}[\text{average slot time}]}. \tag{8.16}$$

Thus, the throughput in a half duplex system is

$$S_{\text{HD}} = \frac{\mathbb{P}_{\text{S,HD}} B}{\mathbb{P}_{\text{S,HD}} L_{\text{S}} + \mathbb{P}_{\text{LC,HD}} L_{\text{LC}}} \tag{8.17}$$

and the throughput of a system with collision detection is

$$S_{\text{CD}} = \frac{\mathbb{P}_{\text{S}} B}{\mathbb{P}_{\text{S}} L_{\text{S}} + \mathbb{P}_{\text{LC}} L_{\text{LC}} + \mathbb{P}_{\text{SC}} L_{\text{SC}} + \mathbb{P}_{\text{FSC}} L_{\text{FSC}}}, \tag{8.18}$$

with $L_{\text{S}} = L_{\text{LC}} = 3.52$ ms equal to the transmission time, $L_{\text{SC}} = L_{\text{FSC}} = 640$ μs equal to the collision detection time and $B = 90$ bytes equal to the number of

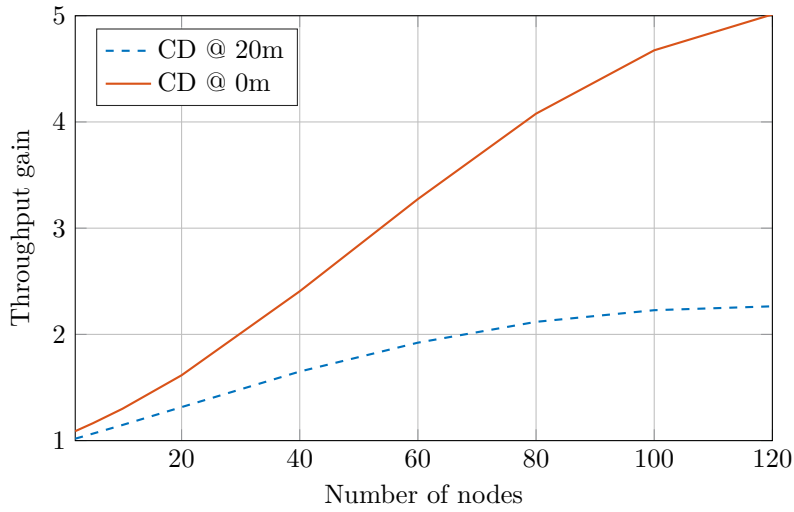


Figure 8.5: System-level throughput gain compared to half duplex.

bits per packet. These formulas are evaluated and shown in Figure 8.5. It is clear that even at longer distances transmitter-based collision detection can improve the throughput. For short distances an improvement of over 5x can be achieved due to the higher probability of detection while for longer distances this is over 2x.

Following a similar rationale that led to (8.17) and (8.18) and using the IBFD energy model from [30], one can calculate the average energy per bit for the different schemes; this is shown in Figure 8.6. Here we see that the energy consumption is lower with collision detection and the benefit of collision detection grows with the number of nodes in the network.

In summary, transmitter-based detection reduces the energy consumption and increases the throughput by vacating the channel sooner when a collision occurs.

8.5 Overview of Detection Techniques

So far, we have discussed the benefits of transmitter-based detection to the network. In this section, we review two different methods for collision and interference detection: the energy detector (ED) and the goodness-of-fit tests. In particular we investigate three different goodness-of-fit tests: the Kolmogorov-

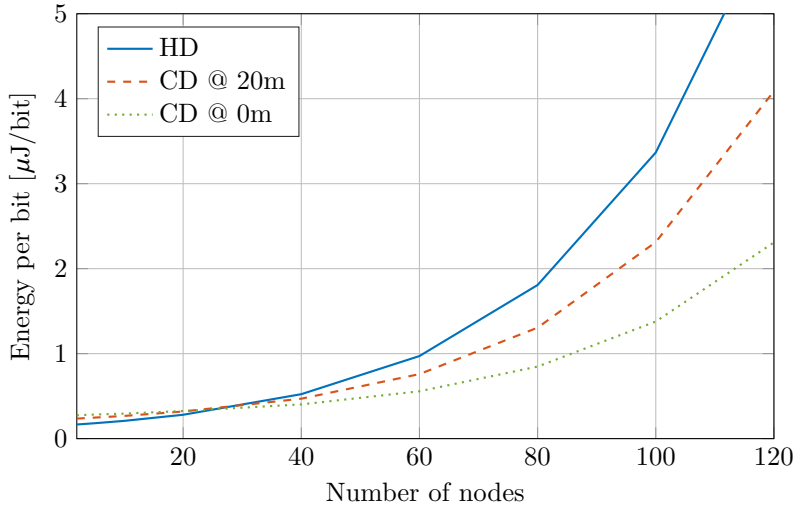


Figure 8.6: System-level energy consumption comparison between half duplex and collision detection.

Smirnov (KS) test [88], the Kuiper (KP) test [74], and the Anderson-Darling (AD) test [95].

8.5.1 The Energy Detector

In this method, the received signal energy is measured and compared to a predetermined threshold to determine whether the interferer is present or not. In particular, the following test is used at the transmitter [97]

$$T_{\mathrm{ED}} = \sum_{n=1}^{N} |Y_n|^2 \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda_{\mathrm{ED}}, \tag{8.19}$$

where λ_{ED} is a predetermined threshold that can be designed to meet a desired performance objective.

The ED is a popular detection technique due to its simplicity, and it merely requires prior information on the self-interfering channel to determine the power of the transmitted signal X_n. However, it has two major drawbacks: the threshold is sensitive to errors in noise floor and remaining SI estimation, and it is unreliable for detecting weak signals. For instance, if the interferer's signal power is close to the noise floor, the source node will frequently confuse between $\mathcal{H}_0$ and $\mathcal{H}_1$ [99].

8.5.2 Goodness-of-Fit Tests

The goodness-of-fit test is a popular approach since it has several nonparametric methods that do not require prior knowledge about the signal to be detected. This is useful in our problem as it allows detecting both collisions and interference. The core idea behind this test is to compare two empirical cumulative distribution functions (CDFs) and determine if they are in agreement or not (this is known as a two-sample goodness-of-fit test). In this work, the transmitter estimates the empirical CDF of the received signal and compares it with the empirical CDF of an interference-free signal (i.e., the empirical CDF of the samples $\{Y_n\}$ under $\mathcal{H}_0$). Mathematically, the empirical CDF of the received signal is computed as

$$\hat{F}_Y(\tau) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(Y_n \leq \tau), \tag{8.20}$$

where $\mathbb{I}(.)$ is the indicator function. Next, we investigate three different variants of this approach.

The Kolmogorov-Smirnov Test

In the KS test, the largest absolute distance between the estimated and the reference distributions is used to determine which hypothesis is true. The transmitter then uses the following test

$$T_{\mathrm{KS}} = \sup_{\tau} |\hat{F}_Y(\tau) - \hat{F}_{Y_0}(\tau)| > \lambda_{\mathrm{KS},\alpha}, \tag{8.21}$$

where $\hat{F}_{Y_0}(\tau)$ is the empirical CDF of the interference-free signal. If $T_{\mathrm{KS}} > \lambda_{\mathrm{KS},\alpha}$, then the hypothesis is rejected with a significance level α.

The advantage of the KS test is that it can be applied to any continuous distribution function without any prior knowledge on the distribution of the received signals. However, this test requires an interference-free period to compute $\hat{F}_{Y_0}(\tau)$. In addition, this test is less sensitive to variations in the tails of the CDF in comparison with variations around the median. This limitation is tackled by the Kuiper and the Anderson-Darling tests.

The Kuiper Test

The Kuiper test can be considered as a variant of the KS test, and it is expressed as

$$T_{\mathrm{KP}} = \sup_{\tau}\{\hat{F}_Y(\tau) - \hat{F}_{Y_0}(\tau)\} + \sup_{\tau}\{\hat{F}_{Y_0}(\tau) - \hat{F}_Y(\tau)\} > \lambda_{\mathrm{KP},\alpha}. \tag{8.22}$$

Similar to the KS test, T_{KP} is compared with a threshold to reject $\mathcal{H}_0$ with a predetermined significance level. Note that by looking at the maximum deviation of the empirical CDF *above* and *below* $\hat{F}_{Y_0}(\tau)$, this test can capture variations in the tails of the CDF, and thus the performance is generally superior to the KS test [100].

The Anderson-Darling Test

This test assigns higher weights on the tail distributions in order to improve the sensitivity to the variations of the extreme points of the distribution. The test is expressed as [101]

$$T_{\mathrm{AD}} = \frac{1}{N^2}\sum_{i=1}^{2N-1}\frac{(2NM_i - Ni)^3}{i(2N-i)} > \lambda_{\mathrm{AD},\alpha}, \tag{8.23}$$

where $M_i = N\hat{F}_Y(\tau_i^*)$ and

$$\tau_i^* = \inf\{\tau | \hat{F}_Y(\tau) + \hat{F}_{Y_0}(\tau) = 2i/N\}. \tag{8.24}$$

8.5.3 Detection Performance Metrics

In the hypothesis testing problems, two commonly used performance measures are the *detection* and *false alarm* probabilities, $\mathbb{P}_{\mathrm{D}}$ and $\mathbb{P}_{\mathrm{F}}$, respectively*. Typically, $\mathbb{P}_{\mathrm{D}}$ is desired to be high, e.g., 95%, in order to reliably detect collisions and interference, whereas $\mathbb{P}_{\mathrm{F}}$ is desired to be low, e.g., 5%, in order to limit unnecessary transmission interruptions.

Mathematically, $\mathbb{P}_{\mathrm{D}}$ and $\mathbb{P}_{\mathrm{F}}$ are, respectively, expressed as

$$\mathbb{P}_{\mathrm{D}}(\lambda) = \mathbb{P}(T \geq \lambda | \mathcal{H}_1), \tag{8.25}$$

$$\mathbb{P}_{\mathrm{F}}(\lambda) = \mathbb{P}(T \geq \lambda | \mathcal{H}_0). \tag{8.26}$$

*Note that these performance metrics are node level metrics and thus not the same as the system-level detection metrics that depend on node distances, discussed in Section 8.3.

For instance, a hypothesis is rejected in the goodness-of-fit test with a significance level α if $\mathbb{P}_{\mathrm{F}}(\lambda) = \alpha$. Using this expression, λ can be computed as $\lambda = \mathbb{P}_{\mathrm{F}}^{-1}(\alpha)$. However, this equality only holds for continuous distributions. It can be shown [102] that in the discontinuous case, this threshold is conservative and thus $\mathbb{P}_{\mathrm{F}}(\lambda) \leq \alpha$. Finding the correct threshold can be done using simulations or measurements. This will be discussed in more details in the next section.

8.6 Prototype Design and Evaluation

In this section we compare the different detection tests based on simulations and measurements using an IBFD prototype. We first present the setups and next the results.

8.6.1 Simulation Setup

The techniques discussed in the previous section are evaluated using MATLAB simulations assuming ideal frequency-flat SI cancellation and no fading. The simulator generates 5 MHz wide QPSK modulated signals similar to 2 MHz wide IEEE 802.15.4 transmissions, which fit in a 5 MHz channel. The same kind of signals will be used for the measurements. To simulate collisions, both the transmitter and interferer transmit QPSK modulated signals. At the end of Section 8.6.3, we evaluate the performance for different modulations in order to simulate generic interferers. The self-interference is attenuated by δ to account for the analog cancellation. The interfering signal is affected by pathloss and sample clock offset. Next, both signals are combined and noise is added. The resulting signal is quantized to 16 bits and passed through the different detection algorithms. This is repeated 2000 times to average out the probabilities.

The parameters of these algorithms are the threshold λ and the number of samples N required. As explained in Section 8.5.3, the threshold for the detectors needs to be precomputed on separate training data, this can be done beforehand and stored on the device. This is done using a binary search algorithm that stops when the target false alarm probability is met. Unless otherwise stated, we fix the target false alarm probability to 5%. This threshold is then used on the test data to determine the detection probabilities. For all the goodness-of-fit tests, the trained threshold does not depend on the self-interference power δ and the number of samples N, which shows that these tests are independent of the SI power and channel, unlike the energy detector where the threshold varies significantly with these parameters.

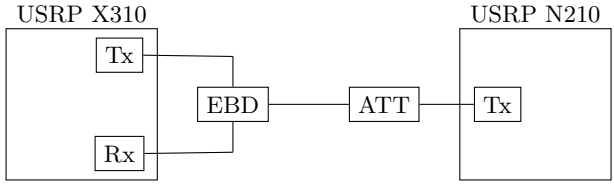


Figure 8.7: In-band full duplex measurement setup using an electrical balance duplexer to cancel the self-interference.

8.6.2 Measurement Setup

Our measurement setup consists of one USRP X310 which acts as the IBFD-capable node and one USRP N210 which acts as the interferer or colliding node. To cancel the self-interference we use an electrical balance duplexer (EBD) [96]. An EBD creates an inverse copy of the self-interference by balancing the antenna impedance. The self-interference and the inverse copy destructively interfere to cancel each other. Figure 8.7 shows how the setup is connected. The transmit port of the first front-end of the USRP X310 is connected to the transmit port of the EBD, while the receive port of the EBD is connected to the receive port of the second front-end of the USRP X310. Next the antenna port of the EBD is connected to a variable attenuator to simulate path loss, which is connected to the transmit port of the USRP N210.

The electrical balance duplexer is first fixed to a self-interference cancellation of $\beta = 50$ dB and then to $\beta = 70$ dB. The measured noise power of the USRP X310 is -90 dBm. The transmit power of both USRPs is fixed to 0 dBm, consistent with the IEEE 802.15.4 standard. The variable attenuator is able to attenuate this signal below the noise floor.

Both USRPs transmit the same IEEE 802.15.4-like signal as described in Section 8.6. This signal is 4 times oversampled both at the transmitter and at the receiver. For each measurement, we decreased the interferer power by 5 dB and carried out 2000 iterations. We first measure the training data, without an interferer present, to estimate the threshold. Next we apply these thresholds to the testing data.

We should note that the threshold of the energy detector is highly dependent on the signal strength. This comes from the fact that the energy detector uses only the present value of the energy while the goodness-of-fit detectors use the difference between two signals. A small difference in the signal strength of the training and testing data therefore results in a high false alarm probability ($> 90\%$). Instead of showing these results we opted to train the threshold for

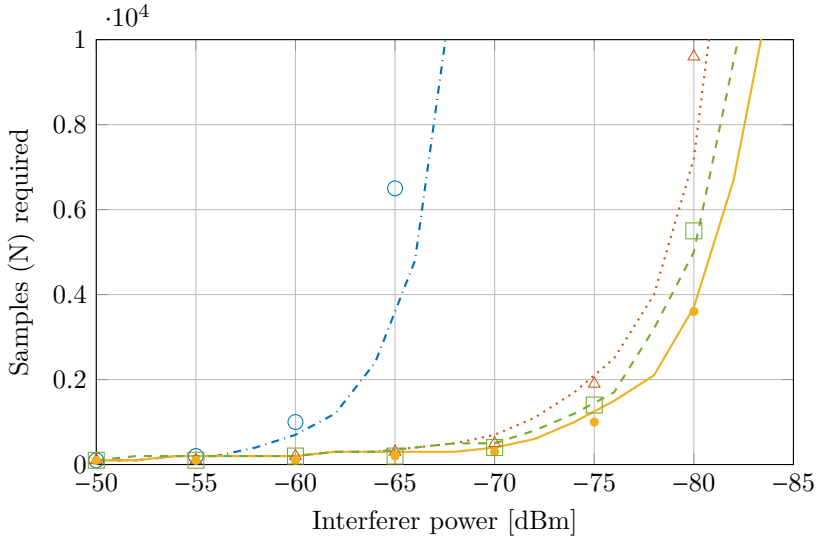


Figure 8.8: The required number of samples to achieve 95% detection and 5% false alarm. Simulation (lines) and measurement (markers) results for KS (···△···), KP (—●—), AD (-□-) and ED (-⊖-). ($\mathbb{E}[|W_n|^2] = -90$ dBm, $\mathbb{E}[|X_n|^2]\beta^{-1} = -50$ dBm)

the energy detector on the testing data instead of the training data. This will give an unrealistic upper bound of what is possible. Such manipulation is not necessary for choosing the thresholds of the goodness-of-fit tests as these are stable under varying signal strength.

8.6.3 Results

To analyze the performance, we first look at the number of samples required to achieve $\mathbb{P}_\mathrm{D} \geq 95\%$ and $\mathbb{P}_\mathrm{F} = 5\%$. Figure 8.8 shows that the number of samples increases exponentially as the interferer power decreases. In other words, to detect weak interferers, very high number of samples is necessary. In the figure, the self-interference is set to -50 dBm ($\delta = 50$ dB) and the noise floor is at -90 dBm. It is clear that the performance of the energy detector is the worst; it achieves a processing gain of $\beta = 15$ dB relative to the self-interference. The three goodness-of-fit tests perform better and each achieves a processing gain of at least $\beta = 25$ dB with the Kuiper test performing the best. To achieve this, the tests need roughly 2000 samples which is equivalent to 4 bytes or 250 μs in an 8 times oversampled IEEE 802.15.4 receiver. It is possible to increase the

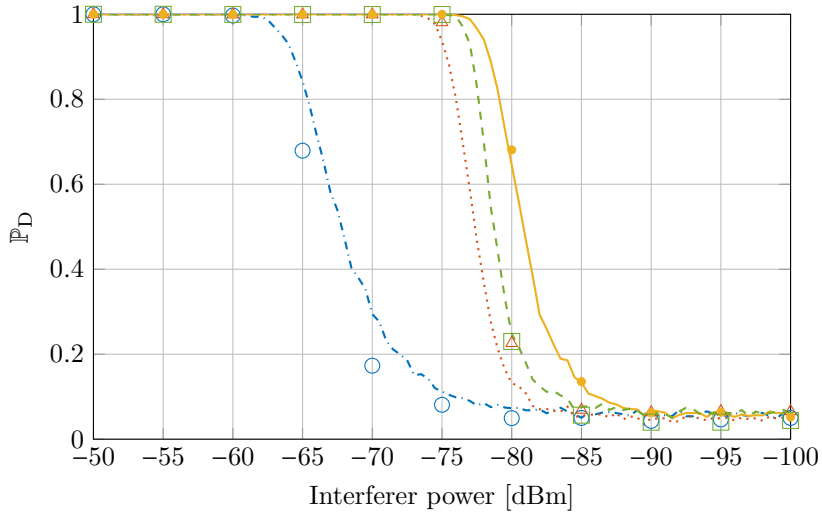


Figure 8.9: Detection probabilities using 2000 samples. Simulation (lines) and measurement (markers) results for KS (···△···), KP (—●—), AD (-⊟-) and ED (-⊖-). ($\mathbb{E}[|W_n|^2] = -90$ dBm, $\mathbb{E}[|X_n|^2]\beta^{-1} = -50$ dBm)

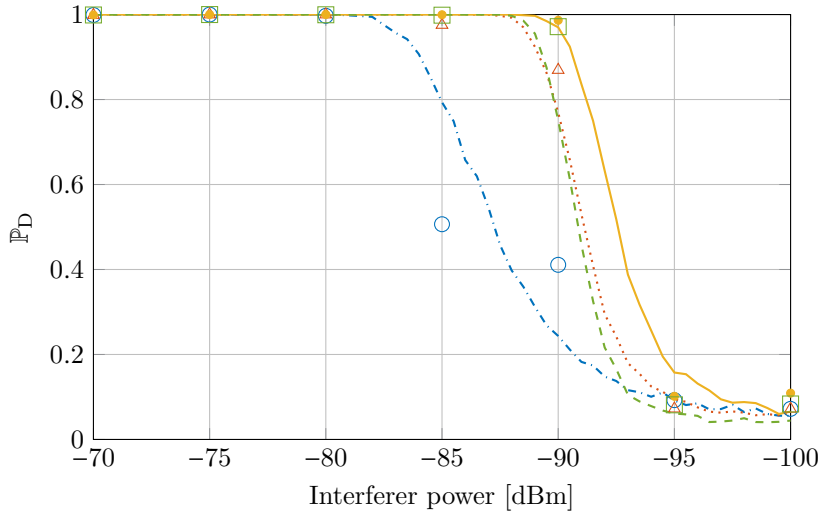


Figure 8.10: Detection probabilities using 2000 samples. Simulation (lines) and measurement (markers) results for KS (···△···), KP (—●—), AD (-⊟-) and ED (-⊖-). ($\mathbb{E}[|W_n|^2] = -90$ dBm, $\mathbb{E}[|X_n|^2]\beta^{-1} = -70$ dBm)

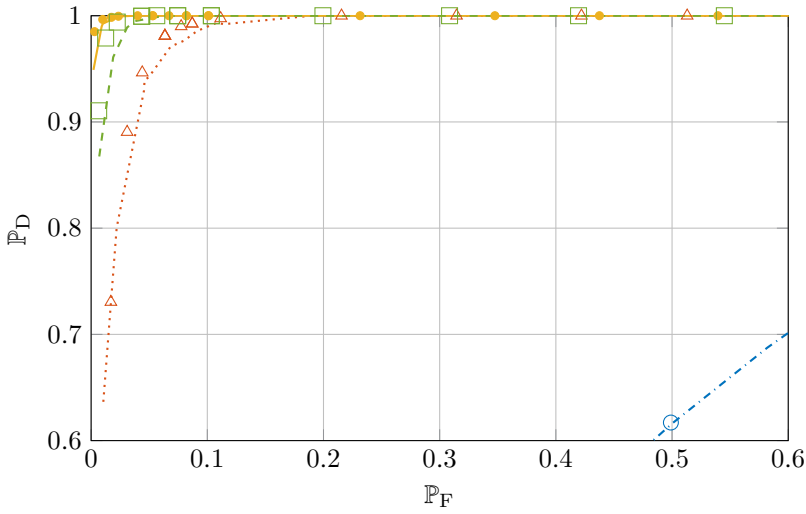


Figure 8.11: Receiver operating characteristic for $\delta = -50$ dBm and $\beta = 25$ dB.Simulation (lines) and measurement (markers) results for KS (···△···), KP (—●—), AD (-⊟-) and ED (-⊖-). ($\mathbb{E}[|W_n|^2] = -90$ dBm)

performance to $\beta > 30$ dB, but this would require more than 10000 samples, which adds delay to the detection. In the remainder of this paper we use 2000 samples to evaluate the detector' performance.

Comparing all detectors for a fixed number of samples (2000) and self-interference at −50 dBm, Figure 8.9 shows again that the performance of the energy detector is at least 10 dB worse than the goodness-of-fit detectors. The false alarm probability is around 5%, independent of the type of detector. Figure 8.10 compares the four detectors but now the remaining self-interference is at −70 dBm ($\delta = 70$ dB). From the previous figure, one would assume the detectors would be able to detect an interferer power of −95 dBm, 25 dB below the SI and 5 dB below the noise floor. However, we see that around −90 dBm, with $\beta = 20$ dB, the performance starts to degrade. The performance here is clearly noise limited, which motivates the assumptions from Section 8.3.2.

In Figure 8.11 we look at the receiver operating characteristic for the drop-off area around $\beta = 25$ dB. Note that we zoomed in for this figure to better distinguish the goodness-of-fit tests. It is clear that the KP test outperforms all the other tests. Even with a false alarm probability of less than 1%, it is able to achieve 90% detection probability. The difference with the other goodness-of-fit tests can be seen more clearly here, the AD test is a close second and the KS test

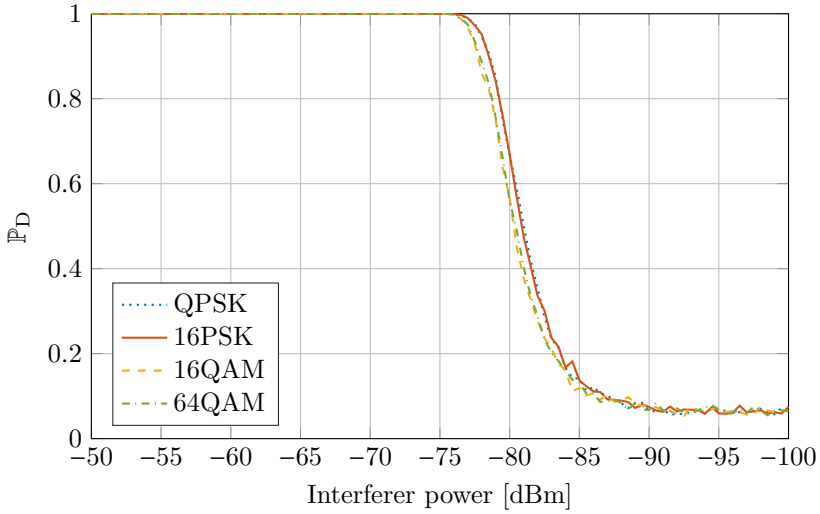


Figure 8.12: Detection probability using the Kuiper test for different modulation types of the interferer. ($\mathbb{E}[|X_n|^2]\beta^{-1} = -50$ dBm, $\mathbb{E}[|W_n|^2] = -90$ dBm)

drops significantly in performance in the low false alarm region. It is evident that the energy detector is inferior to the goodness-of-fit tests.

While the previous results show the performance for detecting collisions of the same modulation type, Figure 8.12 looks at different modulations. This is often the case with interference. In Figure 8.12 we simulate the performance of the Kuiper test as this one has proven to be superior to the other tests. The transmitter is using a QPSK modulation. The difference in performance for the different modulation types is almost indistinguishable, which is to be expected as this test is not using any prior knowledge about the interfering signal.

To summarize, both the simulation and measurement results show that goodness-of-fit tests outperform the energy detector. The measurements results confirm our simulation results that it is possible to detect, with a high detection and low false alarm probability, collisions and interference with a power 25 dB below the remaining self-interference. The Kuiper test has the highest processing gain of $\beta = 25$ dB, while we have shown that the energy detector underperforms even in a best case scenario. Close to the noise floor, the performance is dominated by noise and, therefore, none of the tests were able to detect collisions below the noise floor.

8.7 Conclusions

Future dense communication solutions will have to combine high reliability with low latency and low control overhead. The latter excludes the use of coordination and control messages to schedule users reliably in time or frequency. Contention based mechanisms enable efficient scheduling of multiple users, but fail to give reliability or latency constraints. This paper proposes the use of transmitter-based collision and interference detection using in-band full duplex, enabling the transmitter to detect collisions instantly and vacate the channel as soon as a collision is detected. Our system-level performance analysis shows that such a scheme can increase the throughput fivefold and decrease the energy consumption. The end result is therefore more reliable communication, less interference, and the possibility to instantaneously retransmit packets when a collision occurs.

We show through both simulations and measurements that goodness-of-fit based detectors are viable solutions to detect collisions. Especially the Kuiper based test is able to detect interferers 25 dB below the remaining self-interference but limited by the noise floor. Furthermore, it is possible to do this in less than 250 μs enabling almost instantaneous detection.

Chapter 9

CLAWS: Cross-Layer Adaptable Wireless System enabling full cross-layer experimentation on real-time software defined 802.15.4

Published in:
Bertold Van den Bergh, Tom Vermeulen*, Marian Verhelst and Sofie Pollin, "CLAWS: Cross-Layer Adaptable Wireless System enabling full cross-layer experimentation on real-time software defined 802.15.4" in *EURASIP Journal on Wireless Communications and Networking*, 2014

*Equal contributor

Abstract

Motivation
In recent years, researchers have developed a large and growing set of protocols and algorithms to improve the throughput and capacity of wireless networks. These schemes span the physical (PHY), medium access control (MAC), and higher layers of the protocol stack. Most effective innovations however require cross-layer modifications of both PHY and higher layers. To date, the verification of those designs has been limited to simulations or small setups relying often on off-line processing of the results. MAC layer results that rely on even the tiniest modification of the PHY can only be verified under simplified networking assumptions. Similarly, novel PHY algorithms are typically only verified for a single wireless link, avoiding complex scenarios. Most importantly, there is almost no cooperation between PHY and networking communities, as the tools and testbeds they use are incompatible.

Contributions
In this paper, we propose a methodology for fully flexible PHY, MAC, and network layer verification that is designed to (a) reuse existing software components from PHY and network communities, (b) enable both simple- and expert-level modification and configuration of all components, (c) have real-time performance benchmarked with off-the-shelf systems, and (d) enable large networking experiments including off-the-shelf nodes for rapid experimentation, testing, and comparison. The main contribution of this paper is the introduction of an approach that enables the realization of full software-defined radio (SDR) sensor nodes, all running on a single field-programmable gate array and reusing PHY layer SDR tools and typical operating systems such as Contiki OS. Subsequently, the paper will illustrate the strengths of the proposed approach by demonstrating communication with off-the-shelf sensor nodes. This allows fair benchmarking with state-of-the-art or off-the-shelf solutions. Finally, some cross-layer improvements are proposed and compared with the baseline off-the-shelf system. This proves our claims that the proposed platform is a very useful tool for cross-layer experimentation, in that it allows full cross-layer control of the PHY and network layers, and moreover enables elegant comparison with state-of-the-art designs. This architecture is provided to the open source community (http://claws.be/), in order to become a framework for validating and benchmarking wireless cross-layer innovations.

9.1 Introduction

The flexibility and ubiquity of wireless communication solutions played a very significant role in the tremendous growth of mobile devices such as smartphones and tablet PCs and as such has been an important driver for technology breakthroughs in the last decade. The next, fifth generation of communication solutions (5G) and the Internet-of-Things will require a radical rethinking of the wireless communication landscape to keep improving the spectral and energy efficiency at acceptable cost. State-of-the-art communication solutions already operate close to Shannon capacity. Therefore, in order to support ever-increasing throughput requirements, modern solutions mainly employ two strategies. The first strategy is to use more frequencies which requires increased flexibility at the radio layer, enabled by software radio. The other strategy is to use a more distributed approach; this means that networks are built using many very small cells that cooperate. This, naturally, requires complex networking protocols. Future 5G communication innovations will hence revolve around joint innovation across all layers of the protocol stack, strongly requiring a robust approach for performance evaluation of such complex cross-layer designs. By their very nature, cross-layer communication solutions require mixing different disciplines. As a result, modeling, design, and testing should jointly consider analog, digital, baseband, RF, hardware, and software, resulting in high system complexity even at the level of a single radio. Comparing cross-layer solutions is often achieved by comparing algorithm A on hardware X with algorithm B on hardware Y in non-real-time ideal lab settings. Consequently, performance comparisons are confusing and only hold as far as the calibration methods and assumptions hold. In addition to that, implementation of full communication systems requires a very broad expertise, and it becomes impossible for a single researcher to know sufficiently well all aspects relevant for cross-layer design and implementation.

In this paper, a Cross-Layer Adaptable Wireless System (CLAWS) is proposed that enables gradual improvement and evaluation of cross-layer design innovations. CLAWS is a fully flexible communication node and constructed by combining research tools from both the physical (PHY) and networking community. CLAWS is designed to be user-friendly. Simple experiments can be performed with little knowledge, while experts can still access and modify the core functionality. At the PHY level for example, it is possible to (1) tune parameters of existing functional blocks or (2) add novel functional blocks which requires more experience with the PHY layer (field-programmable gate array (FPGA)) tools. A similar approach can be followed when considering the medium access control (MAC) and network layers, which allow for improved protocol implementations on a default PHY or alternatively take advantage

of the extended PHY functionality. By doing so, the radio can emulate (1) an off-the-shelf radio when standard functional blocks and parameter settings are chosen or (2) a cross-layer improved radio when parameters of existing functional blocks are tuned or novel functionality is added. In this paper, a relevant cross-layer design will be introduced that relies on a novel PHY block for digital mixing or frequency shifting, which is then exploited in user space by a multi-channel MAC. The novel protocol is easily realized in CLAWS, with minimal development effort, and elegantly compared with the single-channel state of the art using the same hardware and network context. This enables effective and correct performance evaluation of the functional improvement, independent from hardware or context calibration errors.

In summary, the proposed methodology in this paper promises to facilitate and benchmark cross-layer radio designs in various large-scale distributed setups. Below, we first detail the state of the art with respect to experimental performance evaluation of wireless solutions. Then, in Section 9.3 the proposed sensor node architecture is introduced. Section 9.4 discusses the performance of our design and Section 9.5 finally presents a small yet relevant cross-layer improvement that could be implemented extremely fast and benchmarked elegantly using our setup.

9.2 State-of-the-Art for user-friendly cross-layer experimentation

To enable realistic and repeatable verification of cross-layer innovations, spanning PHY, MAC, and network layers, it becomes necessary to test the cross-layer improved setup and compare it with an off-the-shelf setup in a similar context. In addition, the methodology should reuse and combine as much as possible research and prototyping tools from PHY and networking communities. Below, we summarize the state of the art with respect to software-defined-radio (SDR) experimentation and experimentation using off-the-shelf radios. Having made this comparison, we illustrate how our methodology wants to improve on that.

9.2.1 Software defined radio approaches

At the core, our methodology relies on SDR [103]. Typical software-defined radios can operate in almost any frequency bands using almost any wireless communication standard. They use a combination of FPGAs, digital signal processors (DSPs), and versatile analog/RF designs to achieve this level of system performance across a range of radio standards. The SDR's core

functionality can be changed by modifying the software and firmware on top of the hardware. Various research groups are embracing the availability of off-the-shelf SDR solutions as a means of showing the ideas and algorithms at work [104]. Many recent innovative ideas have been proven or introduced by means of SDR solutions, e.g., full duplex [105]. These experiments are however, typically, limited to small setups and involve off-line processing of the results, as state-of-the-art SDR approaches do not allow meeting stringent delay requirements and/or often do not implement the full physical layer. To the best of our knowledge, we are not aware of any SDR implementation that allows networking with off-the-shelf radios (allowing to scale up the size of the experiments to hundreds or thousands of nodes at reasonable cost) which requires (a) a real-time PHY implementation and (b) compatibility with common protocol stacks.

Real-time SDR operation is challenging because of processing and data communication delays. In its simplest form, a SDR can be used as an analog front end to convert the radio signal to digital samples. A FPGA then processes the samples further. Most SDRs have a small FPGA that only implements digital filtering and down-conversion. These digital samples are sent over a connection (USB, Ethernet, PCIe, etc.) to a host PC to be processed, which causes long delays (communication delays and processing delays). The measured latencies range from 1 ms up to 30 ms [106]. Obviously, these high latencies limit the response time and precise timing control needed in a MAC design (e.g., the default acknowledgment (ACK) timeout is 48 μs in IEEE 802.11b and 864 μs in the 2.4 GHz IEEE 802.15.4). As such, the communication latency between SDR and host prohibits the development of time-critical MAC solutions.

Nychis et al. [60] present a split functionality for streaming SDR platforms. They have concluded that time-critical radio or MAC functions should not be placed on the host but as close to the radio as possible. They annotate the sample stream with timestamps and control information, allowing them to avoid some, but not all, latency problems. Their design only allows slotted MAC implementations for streaming implementations, as turnaround times cannot be optimized with their approach, making more dynamic networking conditions where nodes contend in real-time impossible. A similar approach is taken by Puschmann et al. [61], where a MAC framework is built on top of the IRIS SDR testbed. While interesting, these approaches only allow experimental verification of some networking scenarios, limited to a small number of expensive SDR nodes and often not in real-time. This limits the possibilities for exploring most non-trivial higher layer protocols.

Bloessl et al. [59] created a GNU Radio-based [52] IEEE 802.15.4 implementation using the USRP N210 hardware. They implement the PHY, MAC, and network layer in GNU Radio which runs on the host PC. This setup requires relatively

cheap RF hardware but offloads all computation to a host computer via a gigabit Ethernet connection. A very powerful computer is required to fully exploit the flexibility. The USRP N210 is able to stream 25 MHz of I and Q samples in both directions, but processing this amount of data is extremely computationally intensive. Moreover, the implementation is not complete as they did not implement carrier sensing and the carrier sense multiple access with collision avoidance (CSMA/CA) protocol.

To address these issues, parts or even all of the processing can be done by the FPGA on the SDR platform. This approach completely mitigates the latency and processing issues but introduces other shortcomings as programming these devices requires a very deep understanding of the underlying hardware and they require domain-specific knowledge (HDL programming, signal processing expertise). As protocol experts often lack knowledge of hardware programming languages or baseband functionalities, it is hard for them to use these platforms or even change the physical layer. While SDR approaches that expose MAC functionality exist, such as WARP, the developed drivers and MAC protocols are platform dependent [107]. Furthermore, these drivers and protocols are provided by the SDR or physical layer community and, since they are often custom non-standard protocols, not widely known to the networking research community. These designs are often considered too limited in functionality for protocol researchers, which are used to working with full protocol stacks, e.g., interoperability with a whole range of off-the-shelf radios is expected, and extended protocol tuning options are desired. Ideally, the SDR should be compatible with existing protocol stacks developed in other communities, enabling to combine innovative PHY designs from the PHY-SDR community with protocol developments from the networking community. This avoids duplicating development work and allows each community to leverage upon its own tools and strengths.

The above constraints limit the use of these SDR platforms towards research on the PHY layer, potentially combined with very simple, not time critical, MAC layers. As such, the basic promise of SDRs (reconfigurable connectivity) is still unobtainable: (a) full open SDR protocol stacks that include MAC protocols are still missing; (b) full operation with complete networking layers (including MAC and routing) is not realizable; and (c) real-time SDR interaction allowing benchmarking with off-the-shelf nodes and the realization of large testbeds has currently not been achieved.

A similar approach as proposed in this paper was proposed in [66], which is a real-time 802.11g PHY and MAC implementation on FPGA. The design focuses on design reusability, and many of the relevant parameters were chosen to be easily configurable. By using Bluespec, user-friendly hardware programming is achieved. A CSMA/CA MAC is implemented on the FPGA, as well as a soft

processor that can possibly run higher layer protocol stacks. While this design is in principle very similar to the method we propose here, they never verified that their design can work with off-the-shelf radios and never ported higher layer protocol stacks on the SDR. In this paper, we provide a full 802.15.4 PHY and MAC layer and also a hardware abstraction layer that allows porting most OS or protocol stacks on our PHY. In addition, we benchmark our design with off-the-shelf radios and realize a relevant cross-layer improvement to illustrate the key benefits of the proposed approach.

9.2.2 Off-the-shelf radio approaches

In addition to PHY layer testbeds, wireless communication test facilities with hundreds of radios exist, such as the ORBIT testbed [108] or the w-iLab.t testbed [109]. The wireless communication nodes consist of off-the-shelf communication solutions such as IEEE 802.11 with little or no flexibility as this is constrained by flexibility offered by the chip implementation. As a result, these testbeds can only be used for higher layer protocol or application research. Key in those approaches is the selection of the appropriate OS and protocol stack for higher layer networking research; often, the challenge is obtaining fast and fine-grained control of the various possible radio chipsets available.

In an effort to solve this challenge, generic flexible MAC approaches, originating from the SDR philosophy, have been proposed, which can be interpreted on the device itself [110, 111]. Based upon the analysis of CSMA, TDMA, and hybrid MAC protocols, the decomposable MAC frameworks define a set of MAC functionalities (blocks) as a library. By combining these blocks using a wiring engine, a wide range of protocols can be realized, nevertheless limited to the predefined MAC blocks in the library. These blocks are however mostly limited by the PHY capabilities, and more powerful MAC innovation would be possible if selected PHY layer implementations would be available such as interference analysis, full duplex or flexible bandwidth, and frequency tuning.

To maximize MAC innovation capabilities, approaches exist that try to expose as much of the PHY flexibility as possible to the user space. This is not straightforward as it requires extensive knowledge of the specific OS, hardware platform, and radio controller implementation. A project that pioneered in the development of a possible solution is the FLAVIA project [112]. The project investigates how to execute MAC and (a subset of) PHY commands without the need to access the firmware of the radio device. This way, the construction of the MAC and some PHY commands can be implemented in highly reusable modules and functionalities. Furthermore, virtualization of radio resources was investigated such that several wireless network stacks could operate in parallel

over the same wireless link. The approach should be extended towards SDR, where more control of the radio is possible, and virtually unlimited MAC design freedom is possible.

Considering sensor networks specifically, an OS for low-power embedded devices is often used to manage wireless devices, using protocols such as IEEE 802.15.4. TinyOS [45] and Contiki [46] are commonly used operating systems for wireless sensor nodes. Many useful and well-known higher layer protocol innovations are already implemented in these operating systems, and the development effort is mainly shifted to the radio-specific control interface. Many developers have made their own set of customizations in order to optimize the control of the radio chipset, but these are obviously not interoperable between chipsets. These customizations are often stored in private repositories and are not made publicly available. In this paper, we show that it is possible to write a generic SDR interface layer that allows running the Contiki SDR stack on top of our SDR PHY.

9.3 The CLAWS architecture

The CLAWS platform enables cross-layer experimentation and benchmarking at various levels of configuration complexity. In this section, we will describe the different modules of CLAWS, starting with a high-level overview of the architecture.

9.3.1 Overview

The architecture of the CLAWS platform is given in Figure 9.1. It is clearly shown how the radio front end, baseband PHY, MAC, and control functionalities are split. The baseband PHY is written in LabVIEW FPGA [58] and implemented on the FPGA to ensure real-time performance. This PHY is interfaced to a MAC processor which forms a bridge between the PHY and a Microblaze FPGA-mapped softcore processor [62]. This generic SDR interface layer provides full control over the PHY, as is needed for cross-layer innovations. In our implementation, we have ported Contiki OS to this Microblaze core in order to provide experimenters with a standard and commonly used network layer environment. Hence, we can leverage all features and innovations provided by Contiki OS. When doing pure PHY layer research, the Microblaze subsystem is not needed and can be disabled. The management application running on the host computer then listens on a transport control protocol (TCP) socket that allows direct control of the radio physical layer by sending low-level control

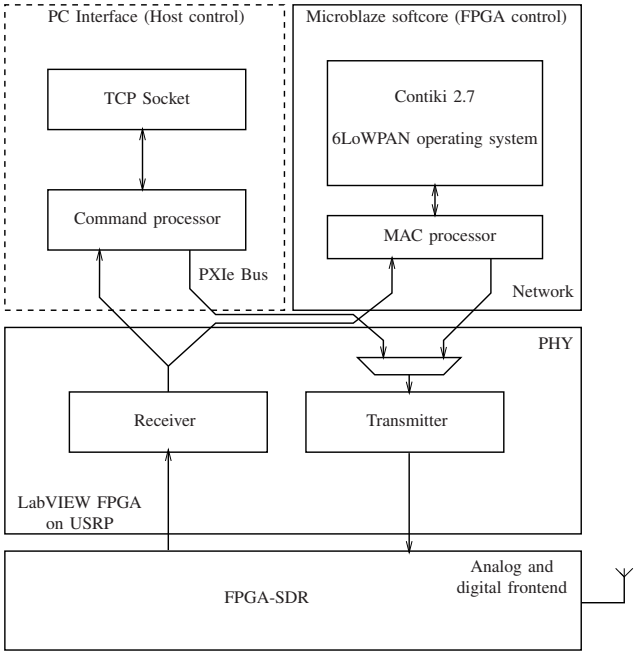


Figure 9.1: CLAWS architecture showing the RF/PHY/networking functionality.

commands through our command processor, as further explained in Section 9.3.5. This host control can be used for automated performance testing of the PHY and link layers when networked tests are not needed.

To build our prototype, we used the NI USRP-2942 [87] which contains a Xilinx Kintex 7 FPGA. The front end has an RF range from 400 MHz to 4.4 GHz and a sample rate of 120 MS/s. The FPGA is connected to the host PC by a PCIe connection. Since our architecture is generic, the hardware is not fixed. It is possible to easily port the PHY layer to any other equivalent NI hardware. All functionality above the PHY, such as the Microblaze subsystem or the MAC processor, can be easily ported to other Xilinx FPGA-based SDRs as they do not depend on LabVIEW FPGA.

9.3.2 User-friendly FPGA-based PHY

Both the transmitter and receiver of our baseband PHY implementation are written in LabVIEW FPGA. The provided functional blocks are parameterized,

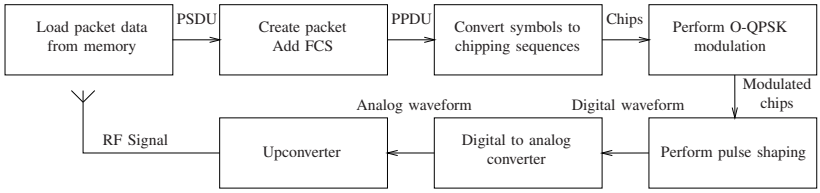


Figure 9.2: Block diagram of the LabVIEW FPGA PHY transmitter.

and these parameters can be tuned by the host or embedded controller without recompilation. In addition, expert upgrades of the PHY are possible, to provide novel or improved functionality, but this requires updates in the LabVIEW FPGA code and recompilation of the design. We will first discuss the transmitter and then the receiver functional blocks.

FPGA PHY transmitter

Figure 9.2 shows the proposed modular PHY transmitter implementation of IEEE 802.15.4. Table 9.1 lists the configuration parameters that are provided to the embedded MAC protocol or to the host. To emphasize the configurability and flexibility of the design, the parameters are split in 'Standard parameters' which are available on most off-the-shelf chipsets and 'Extended parameters' which are provided in CLAWS, as we believed they were relevant for most cross-layer experimentation, beyond the constraints of the IEEE 802.15.4 standard. By relying on the more capable SDR technology, we can allow a range of analog and digital front end parameters, such as the sampling rate of the digital-to-analog converter, oversampling ratio and carrier frequency. These parameters are only bounded by the hardware specifications of the analog front end. By doing so, we can optimize spectrum use by changing channel and bandwidth adaptively.

Our proposed implementation (Figure 9.2) first reads the physical layer service data unit (PSDU) data from memory, which can be a FIFO coming from the host command processor or an internal memory emulated by the MAC processor. Next, the packet data is used to form the PLCP protocol data unit (PPDU) packet, where we can choose to change the start-of-frame delimiter, add a cyclic redundancy check (CRC), change the CRC algorithm or polynomial, and change the maximum length of the packet. The symbol-to-chip mapping block allows us to modify the chipping sequence, which could allow for non-standard spreading codes for, e.g., strengthened privacy. The modulation is performed using offset

Table 9.1: FPGA PHY TX parameters. These are all parameters that can be changed on the CLAWS transmitter.

	Standard parameters	Extended parameters
Read from memory	Data	Origin of data
Form packet	Packet Length Add CRC?	Change CRC Change SFD Maximum packet length
Symbol to chip		Chipping sequence
OQPSK mod		Modulation type
Pulse shaping		Pulse shape
Sampling		Sampling rate
Upconvert	Channel number	Any frequency

quadrature phase-shift keying (OQPSK) with configurable constellation and pulse shape.

In addition to the configuration parameters, it is possible to add functional blocks, such as extra filters for multi-band spectrum aggregation or digital mixers for multi-channel operation. As mentioned before, this requires modifications to the LabVIEW FPGA program and a recompilation of the code. While this is still relatively user friendly because of the graphical user interface, it involves some more advanced PHY layer knowledge, especially when targeting advanced PHY functionality. Gradually, more PHY functional blocks will become available.

FPGA PHY receiver

Figure 9.3 shows the current receiver implementation of IEEE 802.15.4, which is again a library of parameters and functional blocks. Table 9.2 lists all the parameters of the receiver, including both standard compliant modes and extended configuration modes, similarly as for the transmitter. The receiver first downconverts the signal, where a large range of possible sampling frequencies and carrier frequencies is possible, only constrained by the RF front end and analog-to-digital converter. The baseband samples are then used to calculate the received signal strength indicator (RSSI), which is reported to the MAC processor for evaluating the clear channel assessment (CCA). The CCA threshold is parameterized and can hence be controlled by the command processor or embedded controller. The receive datapath then continues first with carrier frequency offset (CFO) compensation, after which the signal is minimum shift keying (MSK) demodulated. The receiver next correlates the demodulated chips

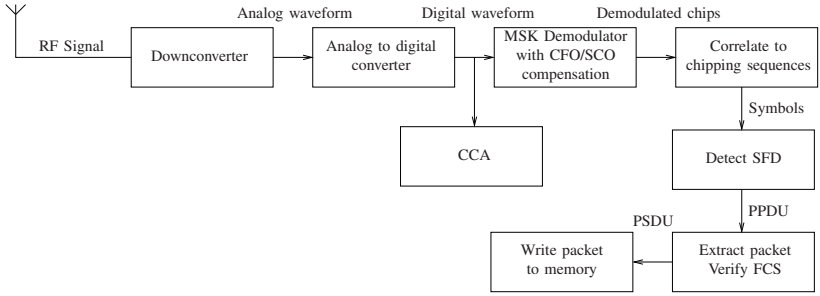


Figure 9.3: Block diagram of the LabVIEW FPGA PHY receiver.

Table 9.2: FPGA PHY RX parameters. These are all parameters that can be changed on the CLAWS receiver.

	Standard parameters	Extended parameters
Downconvert	Channel number	Any frequency
Sampling		Sampling rate
CCA	Limited threshold range	Any threshold
OQPSK demod		Modulation type
Chip to symbol		Chipping sequence
Sync		Other SFD
Extract packet	/	Change CRC Maximum packet length
Write to memory		Change destination

with the chipping sequences to produce the symbols. The system subsequently synchronizes on the start of frame delimiter and extracts the packet. Again, these blocks should be configured to be compatible with the transmitter. For example, if the CRC is changed on the transmitter side, it must be made compliant on the receiver side. The resulting packets are then transferred to either the PC or the embedded system.

Similar to the transmitter, it is also possible to replace or upgrade the available functional blocks, e.g., to add other synchronization or CFO compensation schemes, implement other OQPSK demodulators or add totally novel blocks such as filters for multi-channel bonding, spectrum sensing, or digital mixers for multi-channel operation.

9.3.3 MAC processor

A programmable system has been implemented for converting high-level commands from the network stack, such as 'send a packet', into bit-level instructions for the PHY hardware blocks. This system consists of a small programmable processor dedicated to running the lower MAC tasks. Therefore, it is easy to get sufficiently deterministic real-time performance without impacting the application layers, as would be the case if this was done on the main CPU. The program memory can be updated on-the-fly by the host, which allows for easy switching of the MAC algorithms in use. The interface to the main CPU is by means of a shared memory that both the MAC and main CPU can write to and read from.

To complete the picture, we will explain the tasks performed by the MAC processor when CLAWS is receiving a packet:

- PHY receives a start-of-frame delimiter. The PHY sends an interrupt to the MAC processor, which starts the receive routine.
- PHY receives data bytes. They are transferred to the MAC processor, which stores them in the shared memory. A copy of the packet header is also kept in the local memory of the MAC processor.
- Packet is finished. All bytes have been received, and the MAC processor signals to the main CPU that a complete packet is available in the shared memory. When frame check sequence (FCS) checking is enabled, the signaling is only done if the FCS is correct.
- MAC processor checks the header of the packet to see if an acknowledgement is required. If not, the MAC is now idle again. If it is enabled, it continues with the next steps.
- Start PHY TX. The packet length is three and a FCS should be appended.
- Deliver bytes for transmission. When the PHY asks for a byte to transmit, the MAC will deliver it.
- Transmission ended. The ACK has been sent and the MAC is now idle. An event could be sent to the main CPU if desired.

9.3.4 Networking and OS layer

It is our strong belief that to fully realize the promises of SDR as a wireless innovation platform also for the networking community, it is required to develop

fully embedded systems including software and tools common to the networking community. Most sensor nodes consist of a processor connected to a radio chip and one or more sensors or actuators. To emulate this, our IEEE 802.15.4 radio is complemented with a softcore processor. The chosen softcore processor is the Xilinx Microblaze [62], but alternatives exists, for example, OpenRISC. The Microblaze processor can be programmed in C, which most sensor network researchers will be familiar with since this language is also used for programming the processor in most common off-the-shelf sensor nodes (often a MSP430 or AVR). Furthermore, many sensor node platforms use an operating system to allow smooth development. The standard operating system, Contiki OS, was ported to our platform.

Of course, most real-world applications require more than a system that is just pingable, and many benefits arise from selecting Contiki OS for our platform. First, this allows experimentation with the variation of network layer adaptations that have already been developed for Contiki OS. These innovations can be validated not only on a network of SDRs, but also on a heterogeneous network consisting of off-the-shelf sensor nodes and SDRs. Porting Contiki OS to CLAWS makes operating the system turn-key. Indeed, if a suitable network border router is nearby, one can start CLAWS and obtain 6LoWPAN connectivity immediately, relying on the functionality provided by Contiki OS. Contiki OS also helps with further development by providing a full IPv6 stack that for example allows transmitting sensor values over UDP to a computer or even an embedded webserver. The MAC processor we proposed connects the PHY hardware blocks to the Contiki-based system running on the main processor. Since this MAC processor is separated from Contiki OS, other operating systems, or if needed bare-metal networking code, could be ported to CLAWS by extending the MAC processor where needed. This means that the architecture is not limited to Contiki OS only, yet we believe Contiki OS is a good starting point to tap into the networking communities.

9.3.5 Host control and PHY benchmarking framework

It is understood that not every experiment will require real-time control by the embedded processors, and link or PHY layer experiments, as typically carried out by the PHY community, should still be enabled. To meet this demand, a command interface has been developed. This interface with the host computer is written in LabVIEW. With this interface, the host can transmit and receive packets and control the parameters of the datapath via the PCIe interface. Since the computationally intensive DSP PHY implementation is still executed on the FPGA, the load for the computer is very low. The interface is a standard TCP/IP connection using a command/response protocol, as seen on Figure 9.1.

Exploiting this interface, we have developed a benchmarking framework that can measure bit and packet error rates over arbitrary communication channels. It has been used to compare our PHY implementation against other off-the-shelf PHY solutions. This command/response protocol is generic and can also be implemented on other devices, as has been done for the Zigduino to enable a PHY comparison framework.

9.4 Functional validation and performance results

For our performance evaluation, we will first compare our radio design with off-the-shelf implementations to prove that similar performance can be obtained and that the radio can form a network with those implementations. For PHY testing, we use our PHY benchmarking framework with the host command/control protocol. For network testing, we use the CLAWS nodes with the Contiki OS embedded control in a network. These results will also show that it is very difficult to compare performance of different hardware solutions in exactly the same scenarios, due to the differences in hardware (independent from the functionality). This motivates the need for a universal radio, which can emulate off-the-shelf performance and can hence compare various algorithmic improvements at the PHY/MAC layer, independent of pure hardware specs.

9.4.1 Experimental setup

For our PHY experiments, the setup as shown in Figure 9.4 was used, consisting of the proposed command/control PHY layer benchmarking framework. To ensure PHY layer measurements that can be repeated independently of ambient interference, we connect all nodes with a coaxial cable. In addition to the transmitter and receiver, an external noise generator is added. The noise generator, which is implemented on an additional USRP, is used to create an artificially high noise floor, which allows emulating various signal-to-noise ratios (SNR) without being impacted by receiver sensitivity limits.

We have used commercially available Zigduino nodes [113] to benchmark our implementation. The Zigduino is a board with an Atmel CPU and 802.15.4 radio system-on-chip (ATMEGA128RFA1). These Zigduino nodes run software developed for this project that implements the same command protocol as CLAWS. Since the Zigduino only has a serial port (USB-based, provided via a FTDI UART chip), a TCP-to-serial-port proxy, ser2net, is used. All experiments have been done on channel 26, which corresponds to a frequency of 2480MHz. Transmit power measurements have been performed over a 5 MHz bandwidth.

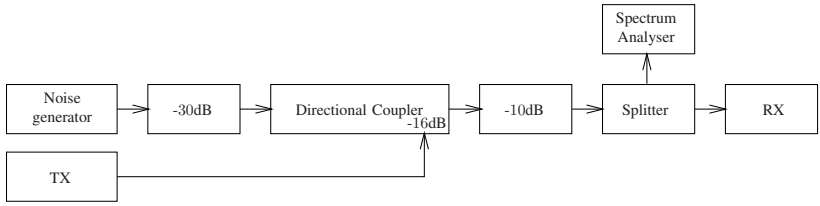


Figure 9.4: Setup used for the BER/SNR experimental results.

One of the advantages of our system is that transmit power control is very linear and over a wide range. The power control of the off-the-shelf radio we used is rather non-linear, requiring the measurements using the Zigduino as transmitter to be done in different output power ranges with attenuators added and removed manually, illustrating how hard it can be to compare various hardware platforms with the same functionality over the same scenario (SNR range in this case).

9.4.2 PHY layer SDR performance

In Figure 9.5, we plot the BER performance comparison between our CLAWS transceiver and the off-the-shelf Zigduino transceiver. It can be seen that a similar BER behavior can be achieved in terms of receiver PHY performance. With similar BER performance, we mean that for high SNR, a similar and low BER is achieved. However, in terms of transmitter PHY performance, we see a 4 dB difference; this is mainly due to signal power measurement errors (as noted before, it was not possible to test the transmitters in the same power range due to limited output power of the Zigduino). In general, a 4 dB performance difference is small enough to test all higher layer improvements. We can conclude that the CLAWS transceiver performance is within the acceptable range of the Zigduino performance, although not identical.

9.4.3 Network layer performance

To evaluate the full system performance, we will measure round-trip delay which is one of the most important parameters. For these measurements, an IPv6 network is set up using a Zigduino as border router (see Figure 9.6). The border router is connected to the host computer via the USB serial port provided by the Zigduino. The USB connection is configured for a bitrate of 1 Mbps. The network was set up to use IPv6 routing protocol for low power and lossy

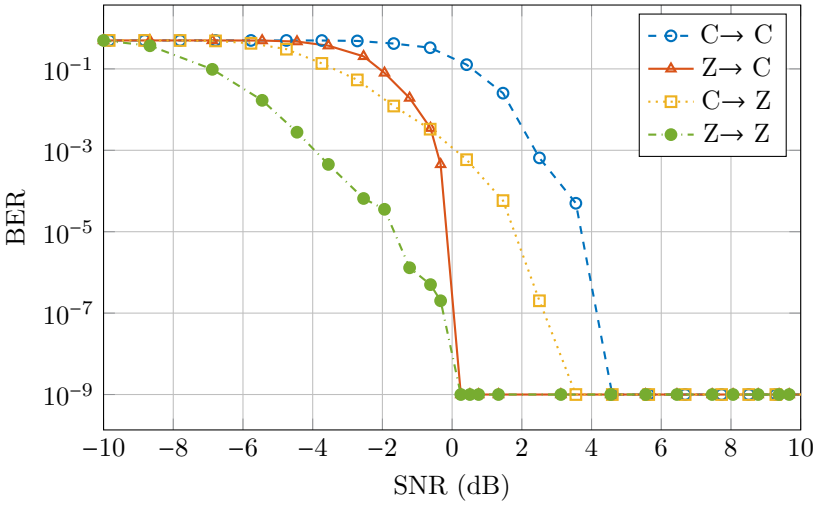


Figure 9.5: Bit error rate comparison between CLAWS (C) and Zigduino (Z).

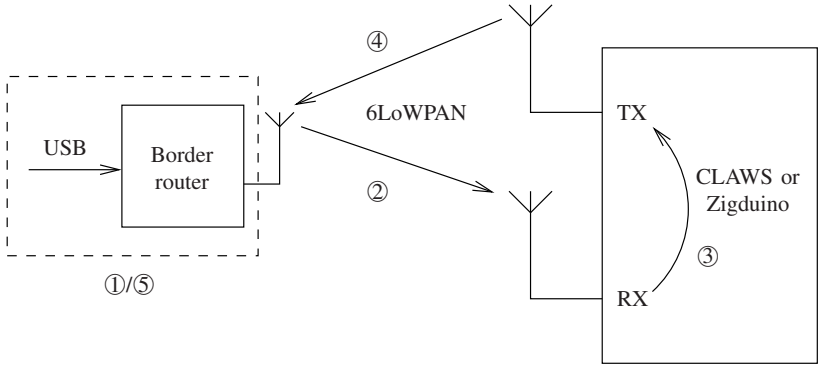


Figure 9.6: Setup used for the network latency tests.

networks (RPL), which is a standard routing protocol designed for low-power wireless networking that is often employed for sensor networks. Of course, a network with a single node is not suitable for evaluating the routing aspects of RPL. This is not the point of this test; we mainly want to demonstrate interoperability with the standard networking technology. To measure the delay, ping packets (ICMP Echo, 37 bytes payload) are sent to and received from CLAWS or Zigduino nodes.

Figure 9.7 shows a timeline of the events that happen during the ping test.

Figure 9.7: Measured ping round-trip time for different targets split into the subparts from Fig. 9.6.

First, the packet has to be processed by the USB connection and border router ①. Next the packet is transmitted over the IEEE 802.15.4 over-the-air channel ②. In the third step, the target device (CLAWS or Zigduino) executes the networking code to produce a reply packet ③. The fourth step is transmission of the reply packet back to the border router ④. Finally, the border router has to process the packet and forward it to the host PC ⑤. The complete round-trip time to the Zigduino is 12.9 ms, while the CLAWS system clocks in at 11.9 ms. Again, we see that the performance of the Zigduino and CLAWS is almost identical. The processing phase of CLAWS is slightly shorter due to the faster CPU. One should note that the radio air-time is longer than what one would expect for 37-byte-long ping packets. This is caused by 6LoWPAN/ICMP protocol overhead, and the downlink packet is 104 bytes, while the uplink packet is 103 bytes long. To demonstrate that RPL routing is also working, we have built a setup consisting of two CLAWS nodes and the border router. Due to the extra hop, the round-trip time to the last node was measured as 19.8 ms; this is not shown in Figure 9.7.

While Contiki OS in the configuration we used for these tests does not use IEEE 802.15.4 ACKs, we have observed the minimum time required by our system to send an ACK. This was significantly faster (8μs) than allowed by the standard (192μs) and serves as an indicator that real-time performance can be achieved with this architecture, potentially for much more demanding protocols, like IEEE 802.11.

9.5 Cross-layer design and benchmarking

From the previous sections, it is clear that the CLAWS performance on both the PHY and MAC layer is comparable to off-the-shelf nodes, making it suited for testing changes in both layers and comparing with the default functionality (on the same hardware, avoiding calibration errors). As indicated in the introduction, we believe most groundbreaking research requires a cross-layer approach. Due to the way CLAWS is built, it is perfectly suited for implementing innovative

solutions requiring changes to be made to more than one layer, leveraging also on tools commonly known to the PHY or networking communities (SDR and Contiki OS). Below, we give an overview of existing cross-layer designs that could benefit from CLAWS. Next, a very simple case study is presented and analyzed. The main goal of this case study is to show the cross-layer experimentation potential of our platform.

9.5.1 Cross-layer design on CLAWS

Various cross-layer design approaches, proposed in literature over the past decade, could easily be verified in CLAWS. In [114], a very scalable networking solution is proposed that allows WiFi networks to adapt to the available spectrum precisely, tuning center frequency and bandwidth. The authors however note that due to the capacity limit of the general-purpose processor, they cannot run their algorithm in real-time and offline decoding is necessary. Such schemes could be implemented on the CLAWS architecture elegantly and would only require some changes to the Contiki protocol layers, taking advantage of the extended parameter set exported to the higher layers.

Alternatively, a large amount of multi-channel MAC protocols have been proposed in the past, but real-time validation has up to now been challenging as fast enough tuning of the PHY channel has been difficult to achieve using off-the-shelf chipsets [115]. With CLAWS, leveraging the wide bandwidth of the RF front end used, it becomes possible to implement multi-channel MAC protocols with direct digital synthesis frequency shifting only. By adding a simple numerically controlled oscillator to the PHY, CLAWS can receive on various channels simultaneously or can receive on channel A and transmit on channel B with virtually no channel switching delays (just some clock cycles for the digital mixer). This is powerful as it allows multi-channel MAC prototyping beyond current radio channel switching limitations. Of course, if this level of performance is not needed, it is also possible to reconfigure the hardware phase locked loop (PLL). In this case, the PLL will need to relock after every frequency change, which reduces throughput and increases latency.

Tytgat et al. [116], for example, argue that no single channel is optimal for a large IEEE 802.15.4 network, and ideally different channels are selected for each packet reception (Receiver Defined Transmission). Performance evaluation of the required channel sensing methods and channel selection methods is very challenging using off-the-shelf IEEE 802.15.4 radios, but can be implemented on CLAWS very elegantly. One could perform the implementation in user space, by just controlling the center frequency parameter or by adding a digital mixer in the PHY layer that does the channel switching instantly. It is even possible

to simultaneously use different transmit and receive frequencies. In the next section, both approaches are discussed and evaluated.

9.5.2 Cross-layer design case study

To show the merit of our design a relevant cross-layer design that involves receiving a packet and transmitting it on an adjacent channel was benchmarked in a multi-hop network. Using this multi-channel and multi-node implementation, we have conducted some experiments to show the benefits. A relay node was designed that receives on channel 11 and transmits on channel 13. First, the PHY was changed to allow shifting the receive and transmit frequencies using numerically controlled oscillators. For this, a novel functional block, i.e., digital shifter, was added to the PHY. This approach allows the receiver and transmitter to work simultaneously on different channels in full duplex. Off-the-shelf nodes can only transmit and receive in half-duplex. Second, changes were made to the MAC layer to allow relaying received traffic on the input frequency to the output frequency. For this, the MAC processor was changed to allow transmission and reception of packet data at the same time. This is an adaptation that is not specified by the 802.15.4 standard. Finally, the networking layer was configured to forward received packets via the second channel. Because of the small scale of the experiment, Contiki was configured to route the traffic in a static way; this meant setting the routing tables accordingly. The necessary changes on all three layers could be implemented in 15 min using the CLAWS design.

To verify operation, we have used the benchmarking framework, as explained in Section 9.3.5. One Zigduino node was configured to transmit on channel 11 and a second one to receive on channel 13. The setup is shown in Figure 9.8. As seen in Figure 9.9, the system is able to receive data on the first frequency while transmitting on the second. The same multi-channel relay could be set up by implementing changes at the MAC layer only, i.e., upgrading the MAC to allow for dynamic control of the PHY center frequency. The latter approach requires however retuning the analog PLL, which involves a delay compared to the more advanced full duplex approach with digital mixing. Both can however be realized in CLAWS very efficiently and flexibly, and it is up to the protocol designer to decide what is necessary for benchmarking the protocol. When implementing the proposed multi-channel relay on off-the-shelf Zigduino nodes, of course, only the latter approach is possible, and the MAC design becomes limited by channel switching delays or hard constraints imposed by the hardware.

To further test this simple case study, we have carried out some throughput experiments. The setup is shown in Figure 9.10. The left Zigduino transmits

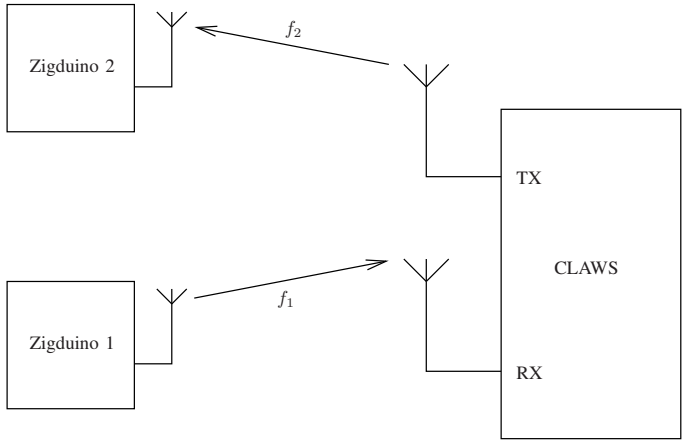


Figure 9.8: Architecture of the cross-layer relay experiment

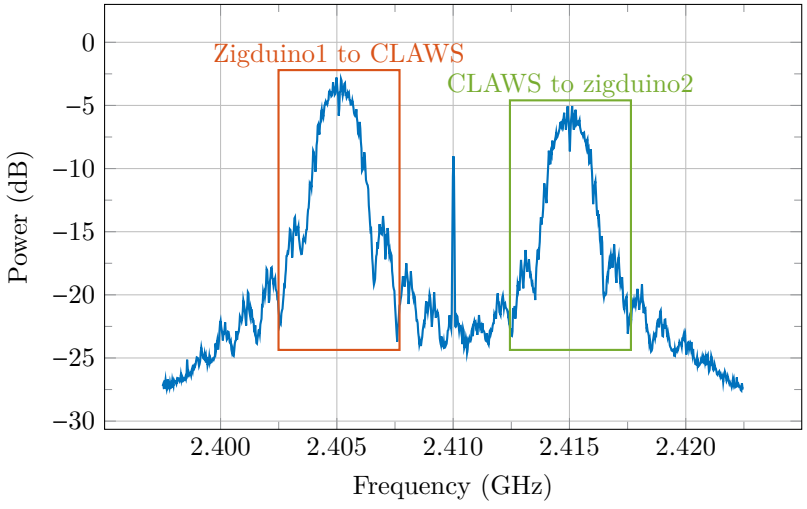


Figure 9.9: Relay design spectrum: The spectrogram shows channels 11 and 13 being used.

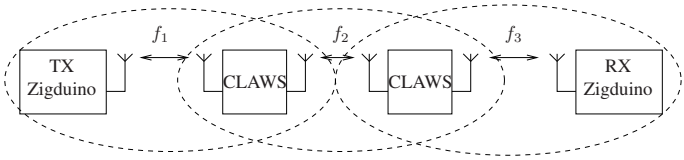


Figure 9.10: Setup used to benchmark the cross-layer multi-channel implementation.

packets whenever the medium is free to the first CLAWS node. The packets are then relayed by this first CLAWS node to the second one. This second CLAWS node again relays the packets to the right Zigduino. As can be seen from Figure 9.10, all nodes are only in range of their neighbors, making multi-hop the only possible path. Three different multi-channel optimizations are benchmarked in terms of throughput against the single-channel, standard compliant implementation. The results are shown in Figure 9.11. In the single channel case, frequencies f_2 and f_3 from Figure 9.10 are equal to frequency f_1. In this case, the throughput decreases a lot because nodes need to wait for the neighboring nodes to stop transmitting. The first improvement enables multiple frequencies for the hops, and a 50% higher throughput is achieved. The second improvement enables full duplex on multiple frequencies; for this, minor changes were needed to the software running on the MAC processor. This second improvement reaches a throughput of around 171 kbps. The 802.15.4 standard mandates a sensing period to determine if the channel is idle. Since in this case we know that no collision can occur, we disable this sensing period. The improvement afforded by disabling the nodes' CCA increases the throughput to 95% of the theoretical throughput.

This simple case study shows two things: First, it shows that it is possible to make changes to multiple layers on CLAWS and benchmark them in real-time. Second, it shows that small cross-layer improvements can have huge benefits.

9.6 Conclusions

This paper describes a cross-layer, flexible SDR solution and benchmarking framework, using commercially available SDR technology. The developed system can communicate with off-the-shelf sensor nodes and allows for tuning of both PHY and higher protocol layers. This architecture is provided to the open source community [67], in order to become a framework for validating and benchmarking wireless cross-layer innovations. Future work involves adding novel functional blocks enabling improved PHY performance, as well as more

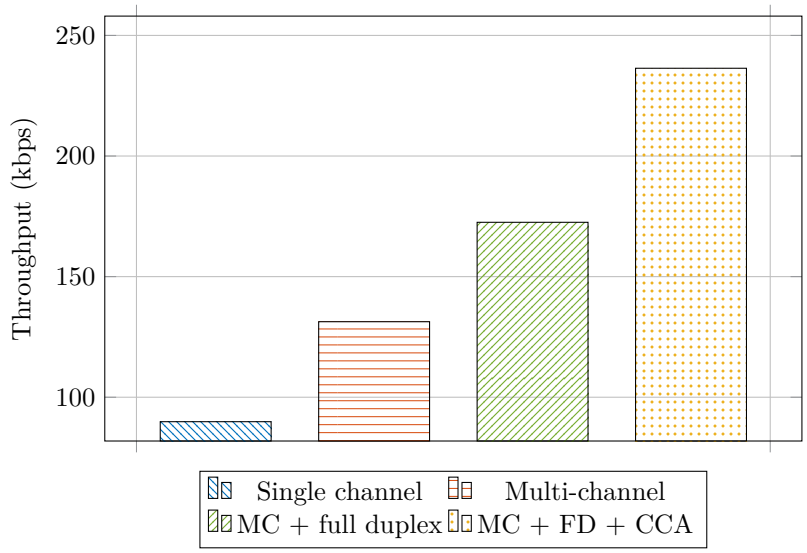


Figure 9.11: Comparison of different multi-channel implementations in terms of throughput.

configuration options or richer protocol development. Also, other technologies such as IEEE 802.11 will be selected and added to the framework. Finally, larger scale benchmarking experiments will be conducted, testing the SDR in large networking configurations, such as the one available in the FP7 project CREW [117].

Author's contributions

Bertold Van den Bergh was responsible for the VHDL programming, MAC processor and Contiki integration, and Tom Vermeulen for the LabVIEW programming. Both cooperated on the experiments and writing the paper.

Chapter 10

Sense and Abort: a Real-Time In-Band Full Duplex Collision Detection Network

Submitted to:
Tom Vermeulen and Sofie Pollin, "Sense and Abort: a Real-Time In-Band Full Duplex Collision Detection Network" in *IEEE International Conference on Computer Communications*, 2018

Abstract

The massive densification of wireless devices increases the number of collisions and interference, which negatively affects throughput and energy consumption. In this paper, we implement and analyze a novel in-band full duplex collision and interference detection scheme, and compare it with traditional half duplex transmissions. Our scheme detects collisions instantaneously at the transmitter and aborts its transmission when a collision is detected. We implement both PHY and MAC on a FPGA-based SDR, enabling real-time wireless collision and interference detection experiments. Our measurement setup consists of six software defined radios with in-band full duplex analog self-interference cancellation, capable of detecting collisions and interference 20 dB under the remaining self-interference level within 400 μs. Our prototype can reach up to 70 dB self-interference cancellation using an electrical balance duplexer. The nodes follow a novel wireless CSMA with near-instantaneous collision detection protocol. Moreover, our SDR networking experiment shows that this novel PHY and MAC scheme improves throughput by 25% already for a small network of 6 nodes, while our simulations show that for a network of 200 nodes, a throughput increase of more than 120% can be achieved.

10.1 Introduction

With the introduction of more wireless technologies within the same crowded spectrum, it becomes critical to ensure that the scarce spectrum is only used for successful transmissions. When a problem occurs due to a collision or interference, a packet transmission should be aborted as soon as possible. While collision detection is typically implemented in wired carrier sense multiple access with collision detection (CSMA/CD), this was believed to be impossible in the wireless context where the collision cannot be heard in the presence of a strong self-transmitted signal, called the self-interference. With the introduction of in-band full duplex (IBFD) [11, 37], it becomes possible to start considering the benefits of collision and interference detection for wireless communication. IBFD is capable of achieving up to 70 dB analog self-interference cancellation even for small CMOS-based RF self-interference cancellation solutions [38].

Current wireless medium access control (MAC) protocols typically use CSMA with collisions avoidance (CA), which waits a random time before transmitting in order to avoid collisions. However, this avoidance has been shown to be ineffective for large networks due to the high probability of a collision [16, 18]. These collisions waste valuable resources such as energy and spectrum. If one

could sense these collisions and abort the ongoing transmission, this would free up the spectrum and make the spectrum utilization more efficient.

As the wireless context is known to be not deterministic, the design of reliable sensing or detection algorithms typically requires sensing over a longer time, to average out the noise or implement other statistical analysis methods. While sensing performance was already studied intensively in the context of cognitive radio [118], it requires more analysis to see how these algorithms can be used for in-band full duplex and if they are sufficiently useful to build realistic wireless sense and abort MAC protocols.

If sufficiently reliable detection algorithms can be designed, which carefully balance the probability of detection and false alarm, the critical question remains if they can be implemented fast enough to enable abortion of the transmitted signal, enabling to retransmit faster and achieve a higher network throughput. Moreover, such novel ideas should be evaluated at network level to confirm the performance improvements empirically in realistic scenarios. The network-level evaluation of in-band full duplex has up until now been missing in the state of the art.

In this paper we provide answers to the following three challenges. (1) Is it possible to reliably sense collisions and interference during a transmission? (2) Can this detection algorithm be implemented to run in real-time on a software defined radio? (3) What is the performance improvements of such an algorithm in a network of multiple nodes?

The techniques developed in this paper can be applied on any CSMA/CA based system, however in this paper we focus on the IEEE 802.15.4 standard [12]. The standard enables low power communication for Internet-of-Things applications. Moreover, the data rate is lower than WiFi-based systems which relaxes the requirements from a proof-of-concept point-of-view. It is also a very relevant technology for dense and reliable IoT applications, as IEEE 802.15.4 is an excellent candidate for smart automation and the industrial Internet-of-Things.

The main contribution of this paper is the design and analysis of a near-instantaneous collision and interference detection solution for in-band full duplex. More specifically, this paper has the following three contributions.

- **Reliable collision detection:** Our algorithm is designed to carefully balance detection and false alarm rates, and we show that it can reliably detect collisions and interference, 20 dB below the remaining self-interference, with a detection probability of 100% and a false alarm rate of 5%.

- **Near-instantaneous detection implementation:** The collision detection algorithm is implemented on the FPGA of the USRP [87] to enable collision and interference detection within 10% of a typical IEEE 802.15.4 packet. Our analysis shows that this near-instantaneous implementation achieves 72% of the maximum throughput that would be achieved when collision waste is minimal, as achieved by an ideal instantaneous and perfect collision detection implementation.
- **Network of 6 USRP prototypes:** The algorithm is shown to work sufficiently reliable and real-time in a network of 6 USRPs consisting of nodes running our novel sense and abort physical layer (PHY) and MAC protocol. A throughput gain of 25% is achieved, for the detection delay achieved in the proof-of-concept, and for a distance between the multiple transmitters of around 1 m. Moreover, for comparison purposes, we implemented the state-of-the-art half duplex CSMA/CA protocol in real-time on 25 USRPs.

To the best of our knowledge, this is the first proof-of-concept of in-band full duplex devices running in a network, showing potential networking gains of more than a factor 2 for in-band full duplex, and realized in real-time including a full PHY and MAC implementation.

In the next section, we provide the functional elements that are required for reliable collision detection at the transmitter. Sections 10.3 and 10.4 present and evaluate the near-instantaneous implementation of our sense and abort prototype. Next, in Section 10.5, we run real-time measurements using a network of 6 USRPs with our sense and abort algorithm and 25 USRPs running the regular CSMA/CA protocol. Sections 10.6 and 10.7 present some related work and a discussion. Finally, in Section 10.8 we present our conclusions.

10.2 Achieving collision detection

Enabling instantaneous wireless collision and interference detection requires cross-layer changes to the PHY and MAC layer. We start with explaining the wireless collision detection MAC protocol, and then zoom in on the PHY layer trade-offs between detection performance and delay.

10.2.1 MAC Layer

Starting at the MAC layer, we propose a wireless carrier sense multiple access with collision detection (CSMA/CD) MAC protocol, shown in Figure 10.1. Here

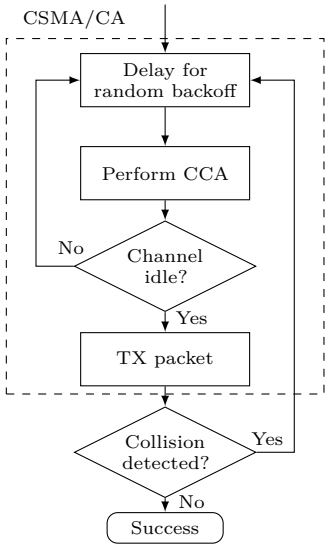


Figure 10.1: The wireless CSMA/CD MAC protocol using in-band full duplex.

the MAC first follows the CSMA/CA protocol but then enables the collision detection functionality as soon as it enters the transmission state. If a collision is detected it will instantaneously abort the transmission and return back to the random backoff state to retransmit the packet.

Recently, several other research groups have analyzed IBFD for collision and interference detection in simulations [17, 98]. The authors in [17] show that collision detection using IBFD can increase the energy efficiency in large networks or allow 50% more nodes in the network for the same energy consumption. In [98] it is proven that collision detection for WiFi can achieve around 90% of the physical layer throughput for 20 nodes compared to 60% for the MAC protocol without collision detection.

To verify these results, we implemented our collision detection MAC protocol in ns-3 on top of the existing IEEE 802.15.4 implementation and compared it to half duplex transmissions and bidirectional full duplex transmissions. The latter being the case were each device involved in the link transmits and receives data at the same time. The ns-3 implementation assumes that it takes at least the header to detect if a collision is present, an assumption which we will validate later in this paper. The throughput gain is shown in Figure 10.2. In an ideal symmetric scenario where each device always has data to send, bidirectional full duplex almost achieves a 100% increase in throughput.

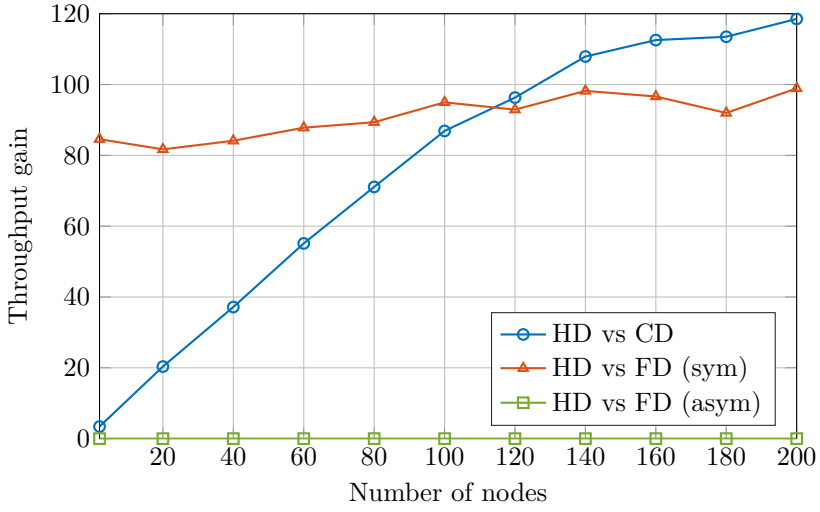


Figure 10.2: Bidirectional full duplex communication is only useful when all devices have data to send. The throughput gain for collision detection is dependent on the network size and thus the number of collisions.

However, in asymmetric traffic conditions where only one device has data to send, the benefit is completely lost. Collision detection on the other hand can gain up to 120% in throughput for very dense networks of more than 200 devices.

Using the in-band full duplex energy model from [30] and the collision information from ns-3, we can determine the energy consumption gains, as shown in Figure 10.3. It is clear that the energy improvement of bidirectional full duplex is small as not only the throughput increases but also the energy consumption. Collision detection on the other hand can save up to 70% in dense networks in terms of energy per bit.

One could also define the average network throughput as the inverse of the average transmission time of a packet, i.e,

$$\mathcal{T} = \frac{1}{T_\mathrm{b} + T_\mathrm{c}\bar{\rho}_\mathrm{c}}, \tag{10.1}$$

where T_b is the time per successfully transmitted bit, T_c is the time per bit of a collision and $\bar{\rho}_\mathrm{c}$ is the number of retransmissions due to collisions. Equation (10.1) ignores the CCA and random backoff parts of the protocol, however, for dense networks this time can be neglected compared to the time the medium is used. For IEEE 802.15.4 networks, $T_\mathrm{b} = 4.26$ μs/bit, which is equal to the time

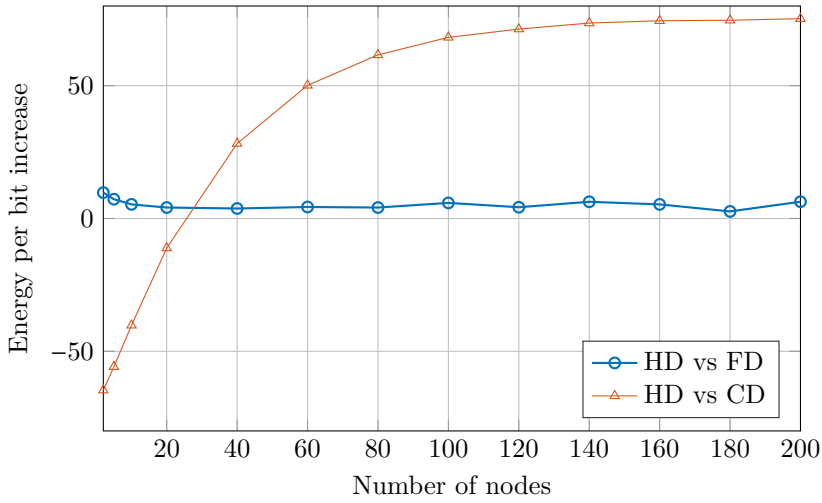


Figure 10.3: Collision detection can save up to 70% in terms of energy compared to half duplex communications.

of the whole packet divided by the number of bits of the payload (120 bytes in our case). For $\bar{\rho}_c$ we ran ns-3 [20] simulations to get an average number of retransmissions due to collisions, this is dependent on the number of nodes in the network. Finally, T_c is dependent on the detection time of the algorithm. Those still familiar with wired hub basics might remember that "To give a chance to A to detect the collision, B does not stop as soon as it detects the collision. This might result in B sending such a short signal that A might ignore it. Instead, B sends a 'jam' signal, long enough to have the energy required for A to notice it." [119]. Therefore, we assume the best case detection time to be the time to receive 1 byte.

Figure 10.4 compares the throughput of the MAC protocol using different detection times for the collision detection. Half duplex (HD) performs as expected the worst as $T_c = T_b$. Using the best and worst case detection time from our measurements in Section 9.4, the throughput can be increased twofold for 20 nodes and 9-fold for 100 nodes. Comparing the area under the curves with the ideal throughput, we show that our detection algorithm can achieve between 62% and 72% of the throughput of an ideal detection algorithm, while for half duplex, this is only 34%.

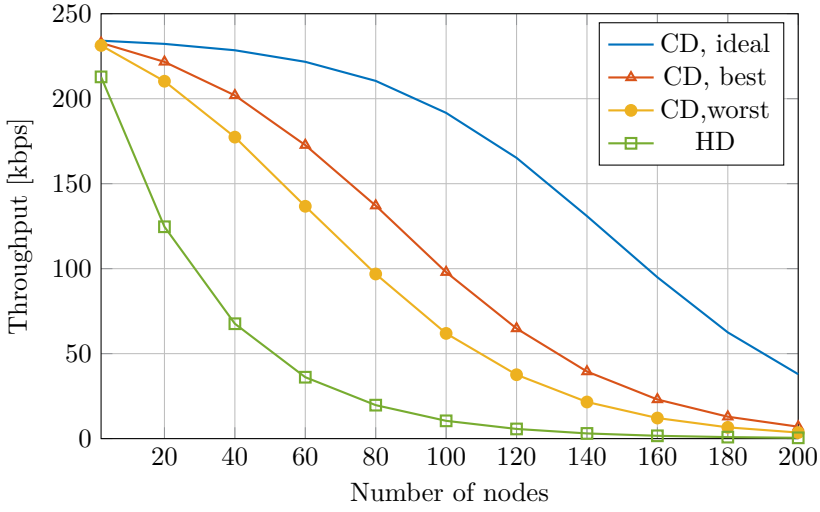


Figure 10.4: The throughput analysis using Eq. (10.4) shows that our collision detection implementation allows to achieve up to 72% of the ideal CD throughput, by comparing the area under the curves of the implemented CD and the ideal CD.

10.2.2 In-Band Full Duplex PHY

On the physical layer, the first step to enable collision detection is to cancel the self-transmitted signal. For this we use in-band full duplex [35, 11, 37], which cancels the self-transmitted signal which leaks into the receiver, allowing both transmitter and receiver chains to operate on the same frequency at the same time. Canceling the self-transmitted signal, also known as the self-interference (SI), is typically done at multiple stages, as shown in Figure 10.5, to allow the signal to be suppressed sufficiently.

First the self-interference is canceled in the analog domain, this can either be at RF or baseband. Multiple research groups have proven the feasibility of these techniques. In [11] the SI signal is canceled by taking a copy of the transmitted signal and passing it through a circuit with varying delays and attenuators. This copy is then subtracted at the receiver side. At baseband the same technique can be applied to cancel the self-interference further [40]. Ideally, most of the cancellation should be done before the analog-to-digital converter (ADC) to enable a higher dynamic range on the received signal and relax the constraints of the ADC. Typical analog cancellation can cancel between 40 to 70 dB of the self-interference [11, 40, 38].

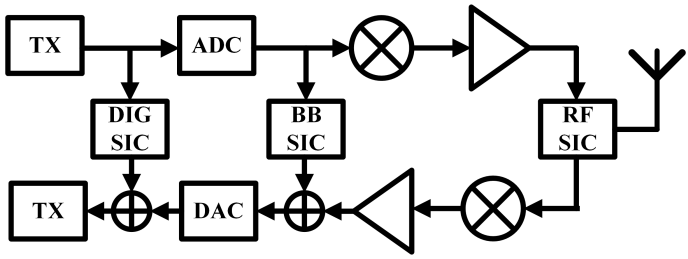


Figure 10.5: Block diagram of the different stages of self-interference cancellation (SIC).

If necessary, further cancellation can be achieved in the digital domain. This is done by using the transmitted samples and estimating the remaining self-interference channel. Again, several research groups have shown techniques to do this [11, 37]. The most comprehensive is presented in [11] where not only the linear terms but also the non-linear terms of the channel are estimated. This technique allows to further cancel the self-interference up to 50 dB.

10.2.3 Collision and interference detection

The final step towards detecting collisions and interference is the classification of the incoming samples to identify a collision. Detecting an unknown signal is hard as no signal features can be used for the detection. In an ideal system, one could simply use an energy detector to distinguish between the collision and just random noise. However, the authors in [99] have shown that due to the uncertainty in noise, there exists an SNR wall below which detection is impossible. The problem is made worse by the presence of the remaining self-interference. Also in our measurements, cfr. Section 10.4, we found the energy detector to perform badly.

In our prototype we use analog cancellation at RF and rely on digital signal processing to detect collisions and interference below the remaining self-interference. The problem can be formulated as a binary hypothesis problem, where the incoming samples (Y_n) are either pure self-interference or a combination of self-interference and a collision, i.e.,

$$Y_n = \begin{cases} h_{\mathrm{SI}} X_n + W_n & \text{if } \mathcal{H}_0 \\ h_{\mathrm{SI}} X_n + h_{\mathrm{I}} Z_n + W_n & \text{if } \mathcal{H}_1. \end{cases} \tag{10.2}$$

Here, h_{SI} is the remaining self-interference channel after analog cancellation, X_n is the transmitted signal and W_n is noise. In the alternative hypothesis ($\mathcal{H}_1$),

h_I is the signal from the interferer to the transmitter and Z_n is the transmitted signal by the interferer.

To detect collisions we are interested in determining to which hypothesis the received samples belong. To determine this, we use a two-sample goodness-of-fit test called the Kuiper test [74], which compares two empirical cumulative distribution functions (ECDF). In [72] it is shown that this test outperforms other goodness-of-fit tests in terms of collision detection performance. The main benefit of using a goodness-of-fit test for the detection is that it does not require any prior knowledge of the signal we are trying to detect. We are only interested in checking whether the received samples are from a known distribution, i.e. our own transmitted samples, or not. Mathematically, the empirical CDF of the received signal is computed as

$$\hat{F}_Y(t) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}(Y_n \leq t), \tag{10.3}$$

where $\mathbb{I}(.)$ is the indicator function, N is the sum of the number of samples in Y_1 and Y_2 and $t \in [Y_1, Y_2]$.

The Kuiper test then calculates the maximum distance between the two ECDFs and compares this to a threshold,

$$\begin{aligned} T_{\mathrm{KP}} = &\sup_{\tau}\{\hat{F}_Y(\tau) - \hat{F}_{Y_0}(\tau)\} \\ &+ \sup_{\tau}\{\hat{F}_{Y_0}(\tau) - \hat{F}_Y(\tau)\} > \lambda_{\mathrm{KP},\alpha}. \end{aligned} \tag{10.4}$$

If $T_{\mathrm{KP}} > \lambda_{\mathrm{KP},\alpha}$, then the null hypothesis is rejected with a false alarm probability α. This means $\lambda_{\mathrm{KP},\alpha}$ should be trained in advance or during idle periods on collision-free samples in order to achieve a false alarm probability of α. From statistical detection theory, we know that we can decrease the uncertainty of our detection by taking more samples into the test statistic or CDF. There is hence a trade-off between sampling delay, implementation and processing delay, and detection performance.

10.3 Sense and abort prototype

Our IBFD prototype is shown in Figure 10.6. The prototype consist of 3 main features, (1) an analog SI cancellation circuit, explained in Section 10.3.1, (2) the IEEE 802.15.4 PHY and MAC layer, explained in Section 10.3.2, and (3) the collision detection algorithm, explained in Section 10.3.3. All digital parts of the prototype are built on a NI USRP RIO [87] software defined radio.

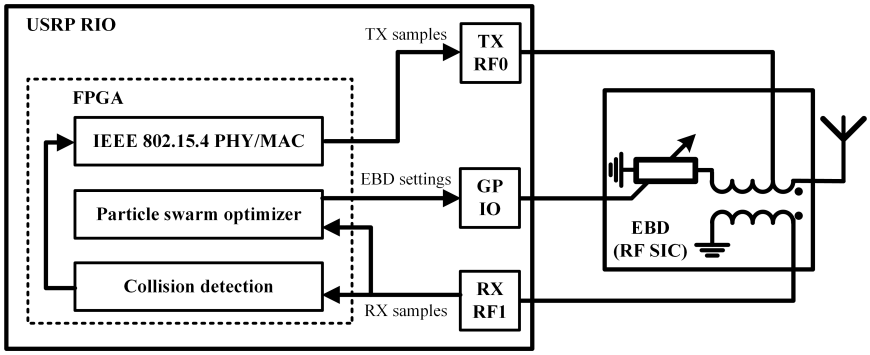


Figure 10.6: Hardware overview showing the USRP and EBD.

Table 10.1: FPGA usage of the full transceiver prototype.

	Usage	Percentage
Register	93921	18
DSP48	168	11
Block RAM	148	19
LUT	92559	36
Slices	31932	50

The FPGA usage of the full transceiver is detailed in Table 10.1. This includes two Xilinx MicroBlaze [62] softcores, one for controlling the self-interference cancellation, and one for running the MAC protocol. Some blocks have been optimized for speed, however, there is still room to optimize for area if necessary. For now, only 50% of the FPGA is used with most logic running at a clock rate of at least 120 MHz and the softcores running at 150 MHz.

10.3.1 Analog SI cancellation

To cancel the self-interference in the RF domain, we use an electrical balance duplexer (EBD) [38]. In the EBD, two signal paths are created, one part of the transmit signal goes to the antenna, another part goes to the balance network. The reflection from the antenna due to mismatch of the antenna impedance is typically the main component of the self-interference. To cancel this signal we tune the balance network in such a way that the reflection is the inverse of the reflection coming from the antenna. Both these reflections are combined at the receive port and destructively interfere with each other. This destructive interference allows to cancel the self-interference up to 70 dB.

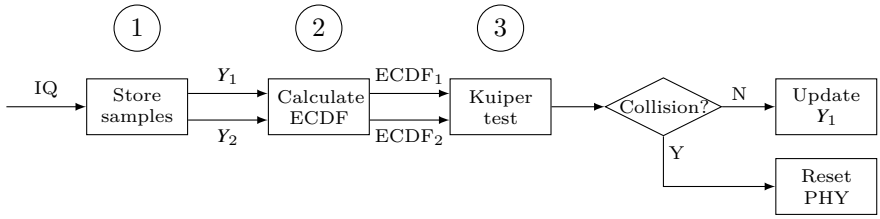


Figure 10.7: The collision detection implementation on the FPGA first calculates the ECDF, which is then used by the Kuiper test to decide whether a collision occurred or not.

The balance network of the EBD consists of four tunable capacitor banks with over 4 billion settings. To find an optimal setting at run-time, a particle swarm optimizer (PSO) is implemented on the FPGA of the USRP [71]. The PSO runs inside a Xilinx MicroBlaze [62] softcore and is connected to the EBD through the GPIO pins of the USRP. This allows us to tune the EBD within 1 ms. The main delay here is the interface with the chip. In our measurements we tune the EBD when the nodes boot up, therefore the 1 ms needed for tuning the EBD does not add extra delay to our collision detection delay.

10.3.2 PHY and MAC implementation

The PHY and MAC layers are based on the IEEE 802.15.4 standard [12] for low power communication. This PHY is implemented on the FPGA using the LabVIEW Communications System Design Suite [64], and runs in real-time at a clock rate of 120 MHz. The PHY is interfaced both with the host for debugging reasons and with a second MicroBlaze running the MAC layer. The full 2.4 GHz PHY is implemented on the FPGA, this includes the clear channel assessment which takes 128 μs according to the IEEE 802.15.4 standard.

The MAC layer, running in the MicroBlaze, is programmed in C and implements the CSMA algorithm. The MAC can send packets to the PHY and start the clear channel assessment (CCA). On top of this, the MAC is enhanced with the ability to enable the collision detection during a transmission. If the collision detection is enabled it will reset the PHY when it detects a collision, without interaction of the MAC, and reports this reset to the MAC.

10.3.3 Collision detection

The collision detection algorithm as detailed in Section 10.2.3 is fully implemented on the FPGA of the USRP and allows to detect collisions within 200 μs. To the best of our knowledge this is the first real-time Kuiper test implementation on a FPGA. Figure 10.7 shows a block diagram of the collision detection algorithm on the FPGA. In a first stage the incoming I and Q samples from the front-end are stored in 27 block RAMs for further parallel processing. 25 of these are used for Z, the combination of Y_1 and Y_2 while one is used to hold the current reference samples Y_1 and the final one is used for Y_2. Next, in a second stage, the ECDFs of the samples in the block RAMs are calculated, this is done in parallel for the samples of the first and the second ECDF. To optimize this algorithm for implementation in hardware, Eq. (10.3) has been split into two parts. The pseudo-code of the state-machine implemented on the ECDF is shown in Algorithm 2. In the first part of Algorithm 2, the 25 block RAMs of Z are read in parallel and compared to Y_1 and Y_2. The output of the first part is used in parallel in the second part of Algorithm 2 as the index of the ECDF memory. At the end of the outer loop, the complete ECDF of Y_1 and Y_2 is computed.

Algorithm 2 Optimized ECDF algorithm

```
for j = 0 to N1 − 1 do
    # First part
    X1 = 0
    Y2 = 0
    for i = 0 to (N1 + N2)/25 − 1 do
        X1 = X1 + ∑{Z(i) ≤ Y1(j)}
        X2 = X2 + ∑{Z(i) ≤ Y2(j)}
    end for
    # Second part
    ECDF1(X1)++
    ECDF2(X2)++
end for
```

In the third stage in Figure 10.7, the Kuiper test uses the CDFs of the previous step and together with Eq. (10.4) decides whether the null hypothesis is rejected or not. This is done by looping over all the elements of ECDF_1 and ECDF_2 and keeping track of the maximum difference between the two. Based on the outcome of the Kuiper test we decide if there is a collision or not. When a collision is present, we reset the PHY and alert the MAC layer. When no collision is present for 10 consecutive evaluations, we update the reference

samples (Y_1), in order to always have the most up-to-date samples and track changes in the reference due to temperature or environmental variations.

The same architecture is used at start-up time to train $\lambda_{\mathrm{KP},\alpha}$. This is done by continuously transmitting a packet and looping over the incoming samples. Next the training logic will use a binary search algorithm to decide which $\lambda_{\mathrm{KP},\alpha}$ has the desired $\mathbb{P}_{\mathrm{FA}}$.

10.4 Measured collision detection performance

To benchmark the detection performance of the collision detection, we plot it against the performance of the energy detector in Figure 10.8. Both these detectors are running on the FPGA of the USRP. The energy detector is implemented using the CCA module of the IEEE 802.15.4 PHY, therefore it takes 128 μs before deciding whether a collision has occurred or not. For each value on the x-axis, we vary the incoming interferer power and each of these values is averaged over 400 runs to smooth out the curves. The figure shows that the detection performance of the Kuiper test is 10 dB better than that of the energy detector, achieving 100% detection probability for signals which are 20 dB below the remaining self-interference. Moreover, the false alarm probability was steady at 5% or less.

The collision detection implementation is able to detect incoming collisions in real-time. Figure 10.9 gives an indication of the overall delay of the detection. If the collision is within the range of detectable interferer powers, it takes the algorithm 306 μs on average. If a collision occurs at the end of the sample storing phase, it can take a worst case delay of 409.28 μs. If on the other hand the collision occurs at the beginning of the sample storing phase, we get a best case delay of 203.28 μs. This is in comparison with the energy detector which, as mentioned above, takes 128 μs but performs worse.

To put the delay in perspective, Figure 10.10 compares the detection delay with an IEEE 802.15.4 packet. In the best case, we can detect the collision just after the PHY header, while in the worst case we would require some more bytes. Note that the time axis is cut at the end to fit the page. In the worst case we would still save around 90% of the transmit time of the whole IEEE 802.15.4 packet. For faster PHYs such as WiFi, the implementation should be optimized a bit further, more on this in Section 10.7.

Looking at the individual delay of the different blocks of the detection algorithm (Figure 10.11), we see that the majority of the delay comes from the ECDF calculation. A small part of the delay is due to the storing of the samples and

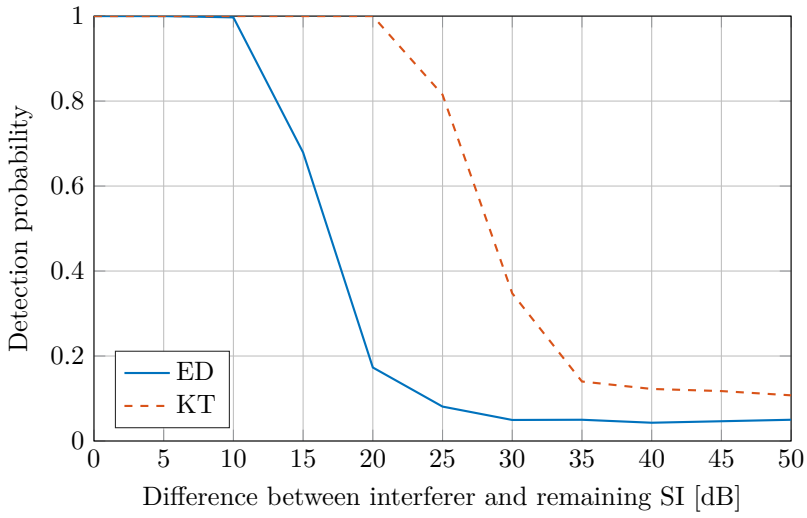


Figure 10.8: The detection probability of the Kuiper test (KT) is more than 10 dB better than that of the energy detector (ED).

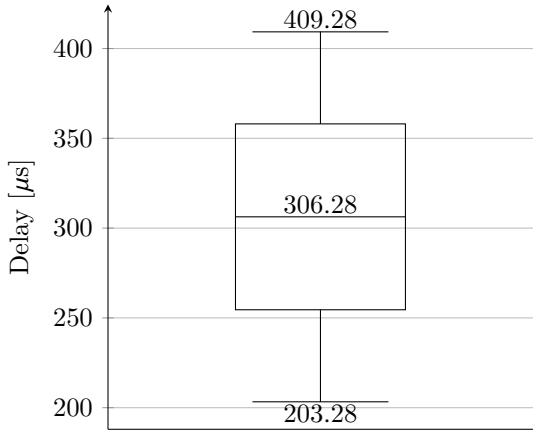


Figure 10.9: The boxplot of the delay of the collision detection algorithm shows that on average it is able to detect collisions within 306 μs.

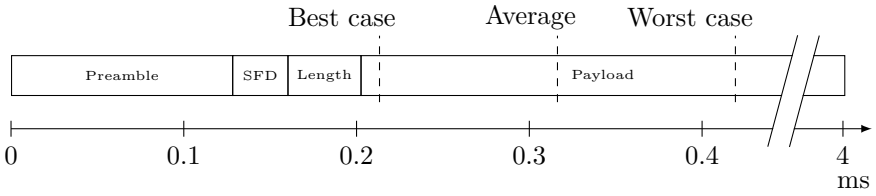


Figure 10.10: Relative to an IEEE 802.15.4 packet, the delay of the collision detection algorithm is comparable to the length of the preamble and PHY header.

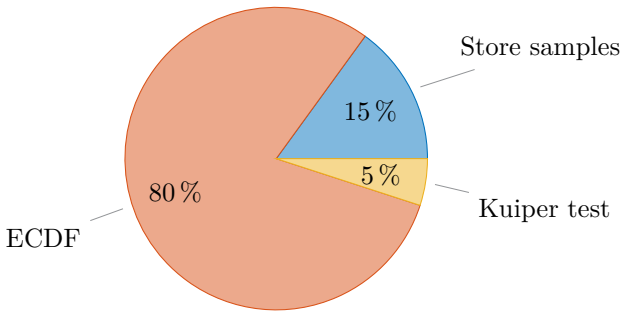


Figure 10.11: Breaking down the delay into its individual components shows that the majority of time is spent in the ECDF calculation.

the computation of the distance between the two ECDFs and the outcome of the Kuiper test. It is possible to reduce the delay of the ECDF by parallelizing Algorithm 2 even further but this requires more space on the FPGA.

10.5 Network Measurements

In our measurements we compare a half duplex network with a network of nodes with collision detection, which is to the best of our knowledge the first experiment with multiple full duplex enhanced nodes interacting in real time in a network. First we discuss the measurement setup in Section 10.5.1, followed by the measurement results in Section 10.5.2.

Figure 10.12: The half duplex setup uses 25 USRPs in a star topology.

10.5.1 Measurement setup

For our half duplex experiments we used 26 NI USRP RIOs mounted in a server rack, as shown in Figure 10.12. This can be considered as a very dense network of 25 nodes configured in a star topology. The 26th USRP is used as central access point. This node logs all the received packets. All the USRPs are connected to one server for automated testing. At the beginning of each measurement, the server will load the bitfiles on each FPGA and configure the frontends accordingly.

The full duplex experiments use a similar setup, although due to the extra room necessary for the electrical balance duplexers, we opted to use external USRPs, this can be seen in Figure 10.13. Here 6 USRPs with EBD are configured in a star topology to transmit to 1 access point. Unfortunately, due to the limited supply of EBDs we were only able to use 6 nodes. The nodes are initialized the same as in the half duplex case. However, some extra configuration is needed for the EBDs, this is done before the actual measurements.

In both cases we fill the packet buffers of all nodes continuously to keep the

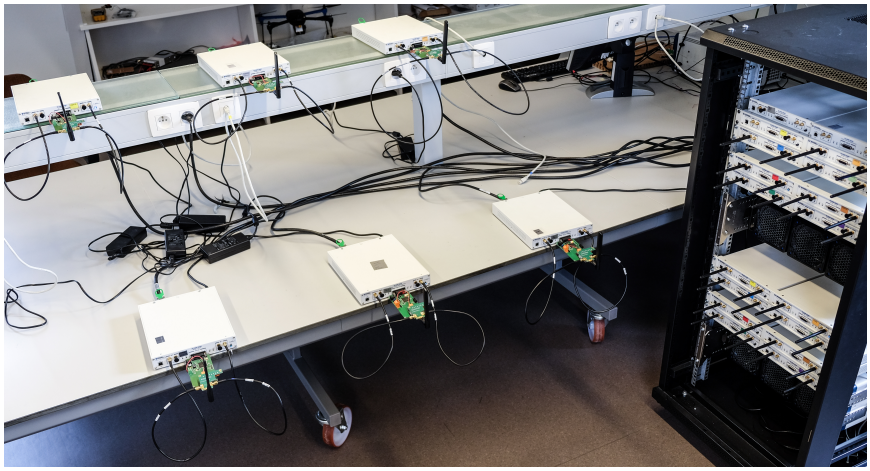


Figure 10.13: The full duplex setup uses 6 USRPs with in-band full duplex capabilities in a star topology.

Table 10.2: PHY and MAC parameters used in the experiments.

Parameter	**Value**
MPDU length	120 bytes
macMinBE	3
macMaxBE	5
macMMaxCSMABackoffs	4

network saturated. Furthermore, the nodes will only transmit uplink data and not request an acknowledgment. Finally, Table 10.2 gives an overview of the PHY and MAC parameters used in the experiments, most of them are default values defined by the IEEE 802.15.4 standard.

10.5.2 Measurement results

Let us first look at the collision probability in function of the number of nodes, shown in Figure 10.14. A typical curve follows an inverted exponential function meaning that the conditional collision probability is increasing monotonously with network density, but ceiling towards 100%. The noisy half duplex measurements follow this curve, moreover, on this data an inverted exponential has been curvefitted, to give an idea how the collision probability will be for an even denser network. As we are measuring a saturated network, the collision

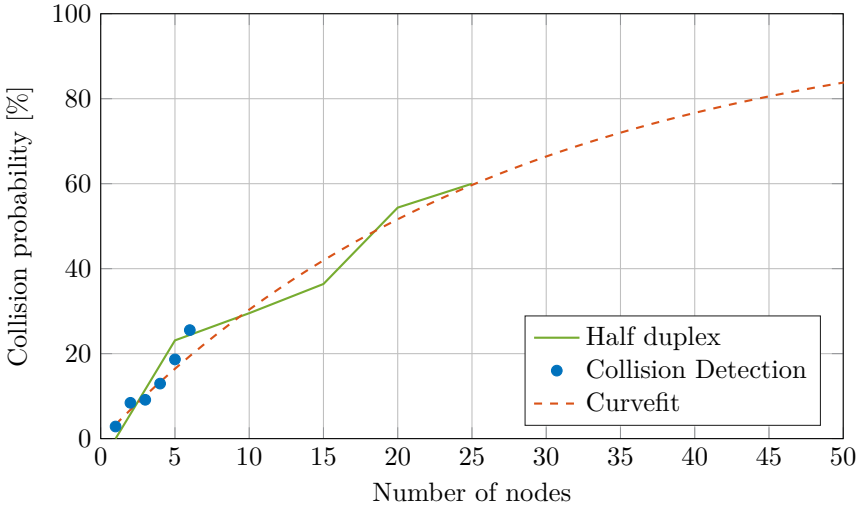


Figure 10.14: The collision probability is identical in saturated traffic conditions for half duplex and collision detection.

probability of the half duplex network and the full duplex network should be similar. In a full duplex network there will be more transmission attempts in a given period, as less time is wasted on collisions, but the conditional collision probability remains unchanged and is only a function of network density. The real gain is not in a reduced collision probability but in a reduced collision time leaving more time for collision-free transmissions. This statement has been validated in Figure 10.14, where it is shown that the collision probability of full duplex is similar to half duplex.

Looking at the throughput, the half duplex network follows a typical curve where the throughput decreases with the number of nodes after a first slight increase with node density due to the decreased network idle time. Our implementation reaches similar throughputs as other measurements [120]. Due to the shorter collision time in the full duplex network, more packets are received, resulting in a 25% increase in overall throughput. With knowledge from Figures 10.2, 10.3 and 10.4, we could extrapolate the CD results, showing that the difference in throughput only increases for a higher number of nodes, exceeding already the factor 2 of bidirectional full duplex for a network of 120 nodes.

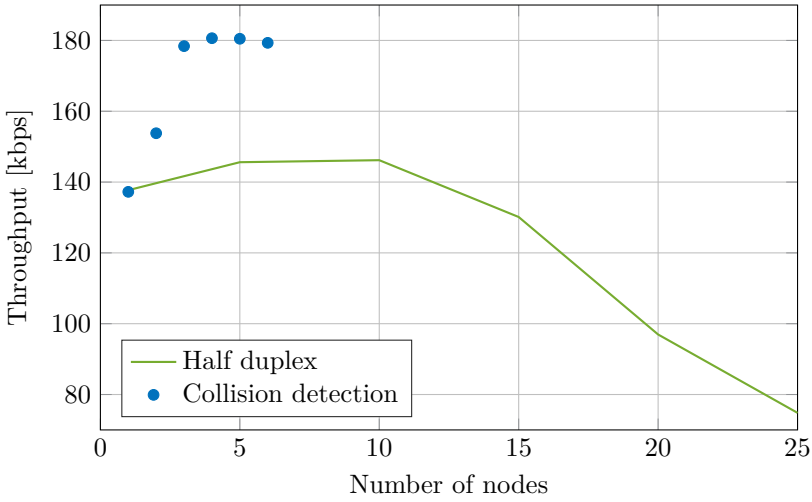


Figure 10.15: The throughput for our sense and abort algorithm is at least 25% higher than the half duplex algorithm for the measured network sizes.

10.6 Related work

To the best of our knowledge, no other works have been published where in-band full duplex collision detection has been prototyped and verified in a network. Most research either focuses on point-to-point links or stick to theoretical evaluations of the possible gains. One such work is published in [17]. The authors verify that collision detection can greatly improve the energy efficiency in dense networks. In [121] the throughput gains of full duplex collision detection for WiFi is investigated. They conclude that throughput gains of up to 40% can be achieved in car-to-car applications with 64 cars.

The authors of [26] present a collision notification algorithm which lets the receiver transmit a notification whenever it thinks it has received a collision. To detect the collisions they use physical layer hints out of the decoder. Their scheme allows to detect most collisions with a probability of detection of 92%. The main problem with this approach is that a feedback channel is necessary to notify the transmitter, making it difficult to take instantaneous action. On top of this, their implementation is non-real-time and requires offline calculations. Moreover, their network measurements are based on a *what if all nodes would have these features* basis.

Research on real-time FPGA-based PHY and MAC implementations is still very

thin. Most implement this functionality on a host computer and try to keep the latency as low as possible [61, 59]. However, this limits the applicability for network testing as often quick reaction times are needed.

10.7 Discussion

What happens if the reference CDF becomes out-of-date? To overcome this problem the reference CDF is updated every time a sequence of 10 collision-free samples are found. This limits the probability of the reference CDF being corrupted with a mis-detection or being out-of-date. However, if the CDF becomes out-of-date, the MAC can reset this reference whenever it decides this is necessary.

How often do you need to retune the electrical balance duplexer? This is dependent on changes in the antenna impedance. In static environments, like our measurement setup, this only needs to be done in the beginning of each test. In more dynamic environments, the higher layers should decide when it is a good time to retune the duplexer. However, if only small changes are required, this can be done instantly during idle moments.

Why not use digital cancellation as well? It is possible to add digital cancellation to our prototype, as there is still 50% space left on the FPGA and all the current blocks would support this. However, our current implementation of using RF self-interference cancellation and the digital processing gain of the Kuiper test allows us to get close to the noise floor of the USRPs. Therefore we opted not to include digital cancellation and save the added processing delay that this would incur.

Does your algorithm work at long distances? The problem of detecting collisions at the transmitter is that not all collisions happening at the receiver can be detected. Moreover, the transmitter will detect certain signals that will not interfere with the receiver, known as the exposed terminal problem. This becomes more problematic at larger distances, as shown in Figure 10.16. By using basic geometry, it is possible to calculate the areas and therefore the detection and false alarm probabilities. The detection probability for a separation of 10 m is around 90%, while at 20 m, this is around 80%. The false alarm probabilities (i.e. exposed terminal detection) are respectively 10% and 20%. These are still respectable giving the high gains collision detection can provide.

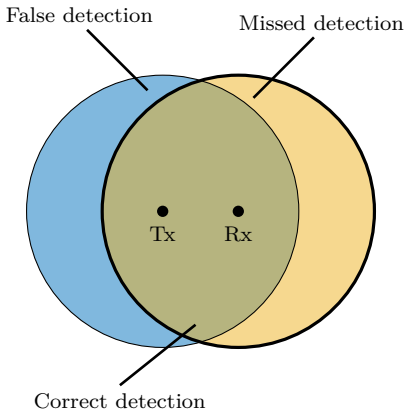


Figure 10.16: As the transmitter moves further away from the receiver, less collisions will be detected.

Adding the detection probability ($\mathbb{P}_{\mathrm{D}}$) and false alarm rate ($\mathbb{P}_{\mathrm{FA}}$) to Equation 7.14 gives,

$$\mathcal{T} = \frac{1}{T_{\mathrm{b}} + T_{\mathrm{c}}\bar{\rho}_{\mathrm{c}}\mathbb{P}_{\mathrm{D}} + T_{\mathrm{b}}\bar{\rho}_{\mathrm{c}}(1-\mathbb{P}_{\mathrm{D}}) + T_{\mathrm{c}}\mathbb{P}_{\mathrm{FA}}}. \tag{10.5}$$

Evaluating this equation for $\mathbb{P}_{\mathrm{D}} = 90\%$ and $\mathbb{P}_{\mathrm{FA}} = 10\%$, only gives a drop in throughput of 0.37% over the full range of nodes that we were looking at in Figure 10.4. While for $\mathbb{P}_{\mathrm{D}} = 80\%$ and $\mathbb{P}_{\mathrm{FA}} = 20\%$, the drop is 0.74%. This comes due to the fact that $\bar{\rho}_{\mathrm{c}}$ becomes dominant in Equation (10.5) for more than 20 nodes.

How will your implementation work with faster PHYs like WiFi? First of all, the sampling rate in WiFi systems is much faster, decreasing the time to acquire and store the samples. Second, the collision detection in the current prototype runs at 120 MHz with a parallelization factor of 25. With proper pipelining it is possible to push the clock higher. Moreover, there is enough space left on the FPGA to parallelize even further. The two combined would enable the prototype to work with faster PHYs like WiFi, which would require sub 1 μs detection times. Ultimately, in a final product the detection algorithm would be put on an application-specific integrated circuit (ASIC) which doesn't have much of the clock and space restraints from the FPGA.

10.8 Conclusions

This paper provides answers to three important challenges. First of all we have shown that reliable detection of collisions and interference at the transmitter is possible, achieving 100% detection probability for interferers which are 20 dB below the remaining self-interference with a false alarm probability of only 5%.

Second, our implementation detects collisions within a detection delay that is lower than 10% of the duration of a typical IEEE 802.15.4 packet transmission proving that instantaneous detection of collisions and interference is possible in wireless communication. Furthermore, our analysis shows that we achieve a throughput that is on average within 72% of the ideal instantaneous collision and interference detection, averaged over a scenario from 1 to 200 nodes.

Finally, we presented and evaluated this implementation using a network of USRPs, each running the IEEE 802.15.4 PHY and MAC with added near-instantaneous collision detection capabilities. For the small network considered, we already show throughput gains of 25%. For a bigger network of more than 30 nodes, our analysis shows that these gains grow to a throughput improvement achieved by in-band full duplex collision detection of more than a factor 2. Hence, in-band full duplex collision detection holds the promise of improving the network throughput beyond what can be achieved with traditional symmetric up- and down-link full duplex.

Bibliography

[1] Global Web Index, "Digital trends in Belgium," visited on 30/04/2017. [Online]. Available: https://pub.be/wp-content/uploads/2015/06/Digital-Trends-in-Belgium-GWI-and-ZO-Infographic-2015.pdf

[2] Business Insider, "Amazon now has 45000 robots in its warehouses," visited on 23/01/2017. [Online]. Available: http://uk.businessinsider.com/amazons-robot-army-has-grown-by-50-2017-1?r=US&IR=T

[3] Google, "Waymo, formerly the google self-driving car project," visited on 25/01/2017. [Online]. Available: https://waymo.com/

[4] Gartner, "Gartner says 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016," visited on 29/04/2017. [Online]. Available: http://www.gartner.com/newsroom/id/3598917

[5] Cisco, "VNI infographic," visited on 26/05/2017. [Online]. Available: http://www.cisco.com/c/m/en_us/solutions/service-provider/vni-complete-forecast/infographic.html

[6] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, March 2014.

[7] Gartner, "Gartner says smart cities will use 1.6 billion connected things in 2016," visited on 26/04/2017. [Online]. Available: http://www.gartner.com/newsroom/id/3175418

[8] Internation Energy Agency, "More data, less energy," visited on 27/04/2017. [Online]. Available: http://www.iea.org/publications/freepublications/publication/MoreData_LessEnergy.pdf

[9] Enerdata, "Electronic domestic consumption," visited on 27/04/2017. [Online]. Available: https://yearbook.enerdata.net/#electricity-domestic-consumption-data-by-region.html

[10] V. Valenta, R. Maršálek, G. Baudoin, M. Villegas, M. Suarez, and F. Robert, "Survey on spectrum utilization in europe: Measurements, analyses and observations," in *2010 Proceedings of the Fifth International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, June 2010, pp. 1–5.

[11] D. Bharadia, E. McMilin, and S. Katti, "Full duplex radios," in *ACM SIGCOMM Computer Communication Review*, vol. 43. ACM, 2013, pp. 375–386.

[12] IEEE, "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4-2011*, pp. 1–314, Sept 2011.

[13] IEEE, "IEEE Standard for Information technology–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE 802.11-2012*, 2012.

[14] Bluetooth SIG, "Bluetooth core specification v 5.0."

[15] S. Pollin, I. Tan, B. Hodge, C. Chun, and A. Bahai, "Harmful coexistence between 802.15.4 and 802.11: A measurement-based study," in *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*, May 2008, pp. 1–6.

[16] S. Pollin, M. Ergen, S. C. Ergen, B. Bougard, L. Van der Perre, I. Moerman, A. Bahai, P. Varaiya, and F. Catthoor, "Performance analysis of slotted carrier sense ieee 802.15. 4 medium access layer," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 9, pp. 3359–3371, 2008.

[17] T. Vermeulen, F. Rosas, M. Verhelst, and S. Pollin, "Performance analysis of in-band full duplex collision and interference detection in dense networks," in *Consumer Communications and Networking Conference (CCNC), 2016 13th Annual IEEE*, 2016, pp. 595–601.

[18] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, 2000.

[19] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part ii–the hidden terminal problem in carrier sense multiple-access and the busy-tone solution," *Communications, IEEE Transactions on*, vol. 23, no. 12, pp. 1417–1433, 1975.

[20] ns-3 project, "Network simulator 3 (ns-3)," visited on 13/01/2017. [Online]. Available: https://www.nsnam.org

[21] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, and W. Dehaene, "Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and improvement perspectives," in *Proceedings of the conference on Design, Automation and Test in Europe-Volume 1*. IEEE Computer Society, 2005, pp. 196–201.

[22] Atheros, "802.11 wireless lan performance," visited on 30/03/2017. [Online]. Available: https://www.security-audit.com/files/atheros_range_whitepaper.pdf

[23] J. Deng and Z. J. Haas, "Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks," in *Universal Personal Communications, 1998. ICUPC'98. IEEE 1998 International Conference on*, vol. 2. IEEE, 1998, pp. 973–977.

[24] J. Peng, L. Cheng, and B. Sikdar, "A wireless MAC protocol with collision detection," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 12, pp. 1357–1369, 2007.

[25] K. Voulgaris, A. Gkelias, I. Ashraf, M. Dohler, and A. H. Aghvami, "Throughput analysis of wireless CSMA/CD for a finite user population," in *IEEE Vehicular Technology Conference*, Sept 2006, pp. 1–5.

[26] S. Sen, R. R. Choudhury, and S. Nelakuditi, "CSMA/CN: Carrier sense multiple access with collision notification," *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 544–556, Apr. 2012. [Online]. Available: http://dx.doi.org/10.1109/TNET.2011.2174461

[27] IEEE, "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," *IEEE Std 802.15.4e-2012*, pp. 1–225, April 2012.

[28] F. Talucci and M. Gerla, "MACA-BI (MACA by invitation). a wireless MAC protocol for high speed ad hoc networking," in *Proceedings of ICUPC 97 - 6th International Conference on Universal Personal Communications*, vol. 2, Oct 1997, pp. 913–917 vol.2.

[29] S. Kumar, V. S. Raghavan, and J. Deng, "Medium access control protocols for ad hoc wireless networks: A survey," *Ad Hoc Networks*, vol. 4, no. 3, pp. 326–358, 2006.

[30] T. Vermeulen and S. Pollin, "Energy-delay analysis of full duplex wireless communication for sensor networks," in *Global Communications Conference (GLOBECOM), 2014 IEEE*, Dec 2014, pp. 455–460.

[31] Texas Instruments, "CC2420 single-chip 2.4 ghz IEEE 802.15.4 compliant and zigbee ready rf transceiver," visited on 20/04/2017. [Online]. Available: http://www.ti.com/product/cc2420

[32] Atmel, "Low power 2.4 ghz transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE, SP100, WirelessHART, and ISM applications," visited on 20/04/2017. [Online]. Available: http://www.atmel.com/images/doc8111.pdf

[33] B. Reynders, T. Vermeulen, F. Rosas, and S. Pollin, "Adaptive in-band full-duplex collision detection for balancing sensing and collision costs," in *2017 European Conference on Networks and Communications (EuCNC)*, June 2017.

[34] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[35] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving single channel, full duplex wireless communication," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 1–12.

[36] M. Chung, M. S. Sim, J. Kim, D. K. Kim, and C. b. Chae, "Prototyping real-time full duplex radios," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 56–63, September 2015.

[37] M. Duarte and A. Sabharwal, "Full-duplex wireless communications using off-the-shelf radios: Feasibility and first results," in *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*. IEEE, 2010, pp. 1558–1562.

[38] B. van Liempd, B. Hershberg, S. Ariumi, K. Raczkowski, K. F. Bink, U. Karthaus, E. Martens, P. Wambacq, and J. Craninckx, "A +70-dBm IIP3 Electrical-Balance Duplexer for Highly Integrated Tunable Front-Ends," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 12, pp. 4274–4286, Dec 2016.

[39] D. J. van den Broek, E. A. M. Klumperink, and B. Nauta, "A self-interference-cancelling receiver for in-band full-duplex wireless with low distortion under cancellation of strong tx leakage," in *Solid- State Circuits Conference - (ISSCC), 2015 IEEE International*, Feb 2015, pp. 1–3.

[40] J. Zhou, N. Reiskarimian, and H. Krishnaswamy, "Receiver with integrated magnetic-free n-path-filter-based non-reciprocal circulator and baseband self-interference cancellation for full-duplex wireless," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan 2016, pp. 178–180.

[41] B. Radunovic, D. Gunawardena, A. Proutiere, N. Signh, V. Balan, and P. Key, "Efficiency and fairness in distributed wireless networks through self-interference cancellation and scheduling," Microsoft Research, Cambridge, UK, Tech. Rep., 2009.

[42] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, "Practical, real-time, full duplex wireless," in *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM, 2011, pp. 301–312.

[43] Texas Instruments, "CC2520 second generation 2.4 ghz ZigBee/IEEE 802.15.4 RF transceiver," visited on 06/04/2017. [Online]. Available: http://www.ti.com/product/CC2520

[44] Sparkfun, "WiFi module - ESP8266," visited on 06/04/2017. [Online]. Available: https://www.sparkfun.com/products/13678

[45] TinyOS, "TinyOS home page," visited on 13/01/2017. [Online]. Available: http://www.tinyos.net

[46] Contiki, "Contiki: The open source operating system for the internet of things," visited on 13/01/2017. [Online]. Available: http://www.contiki-os.org

[47] OpenWSN, "Welcome! - OpenWSN - confluence," visited on 06/04/2017. [Online]. Available: https://openwsn.atlassian.net/wiki/pages/viewpage.action?pageId=688187

[48] IETF, "6LoWPAN status pages," visited on 06/04/2017. [Online]. Available: https://tools.ietf.org/wg/6lowpan/

[49] P. De Mil, B. Jooris, L. Tytgat, J. Hoebeke, I. Moerman, and P. Demeester, "snapMac: A generic MAC/PHY architecture enabling flexible MAC design," *Ad Hoc Networks*, vol. 17, pp. 37–59, 2014.

[50] imec, "imec's technical testing center," visited on 06/04/2017. [Online]. Available: https://www.iminds.be/en/succeed-with-digital-research/go-to-market-testing/technical-testing

[51] National Instruments, "USRP 2922," visited on 06/04/2017. [Online]. Available: http://www.ni.com/nl-be/support/model.usrp-2922.html

[52] GNU Radio, "Wikistart - GNU Radio," visited on 13/01/2017. [Online]. Available: http://www.gnuradio.org

[53] B. Bloessl, "gr-ieee802-11," visited on 06/04/2017. [Online]. Available: https://github.com/bastibl/gr-ieee802-11

[54] ——, "gr-ieee802-15-4," visited on 06/04/2017. [Online]. Available: https://github.com/bastibl/gr-ieee802-15-4

[55] Ettus Research, "RFNoC - RF network on chip," visited on 06/04/2017. [Online]. Available: https://www.ettus.com/sdr-software/detail/rf-network-on-chip

[56] P. D. Sutton, J. Lotze, H. Lahlou, S. A. Fahmy, K. E. Nolan, B. Ozgul, T. W. Rondeau, J. Noguera, and L. E. Doyle, "Iris: an architecture for cognitive radio networking testbeds," *IEEE communications magazine*, vol. 48, no. 9, 2010.

[57] National Instruments, "LabVIEW - national instruments," visited on 06/04/2017. [Online]. Available: http://www.ni.com/labview/

[58] ——, "LabVIEW FPGA module - national instruments," visited on 13/01/2017. [Online]. Available: http://www.ni.com/labview/fpga/

[59] B. Bloessl, C. Leitner, F. Dressler, and C. Sommer, "A GNU Radio-based IEEE 802.15.4 testbed," *12. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze*, p. 37, 2013.

[60] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste, "Enabling MAC protocol implementations on software-defined radios." in *NSDI*, vol. 9, 2009, pp. 91–105.

[61] A. Puschmann, M. A. Kalil, and A. Mitschele-Thiel, "A flexible CSMA based MAC protocol for software defined radios," *Frequenz*, vol. 6, no. 9-10, pp. 261–268, 2012.

[62] Xilinx, "MicroBlaze soft processor," visited on 13/01/2017. [Online]. Available: http://www.xilinx.com/tools/microblaze.htm

[63] ——, "Zynq-7000 all programmable soc," visited on 06/04/2017. [Online]. Available: https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html

[64] National Instruments, "LabVIEW communications system design suite," visited on 24/03/2017. [Online]. Available: http://www.ni.com/labview-communications/

[65] ——, "LabVIEW communications 802.11 application framework," visited on 27/04/2017. [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/en/nid/213084

[66] M. C. Ng, "Airblue: a highly-configurable FPGA-based platform for wireless network research," Ph.D. dissertation, Massachusetts Institute of Technology, 2011.

[67] CLAWS, "CLAWS home page." [Online]. Available: http://www.claws.be/

[68] T. Vermeulen, B. Van den Bergh, and S. Pollin, "Demo: A software defined radio platform for rapid cross-layer prototyping," in *Proceedings of the 2015 Workshop on Software Radio Implementation Forum*, ser. SRIF '15. New York, NY, USA: ACM, 2015, pp. 1–4. [Online]. Available: http://doi.acm.org/10.1145/2801676.2801683

[69] T. Vermeulen, M. Laghate, G. Hattab, B. van Liempd, D. Cabric, and S. Pollin, "Nearly instantaneous collision and interference detection using in-band full duplex," in *Dynamic Spectrum Access Networks (DySPAN), 2017 IEEE International Symposium on*, 2017.

[70] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*, Oct 1995, pp. 39–43.

[71] T. Vermeulen, B. van Liempd, B. Hershberg, and S. Pollin, "Real-time RF self-interference cancellation for in-band full duplex," in *Dynamic Spectrum Access Networks (DySPAN), 2015 IEEE International Symposium on*, Sept 2015, pp. 275–276.

[72] T. Vermeulen, M. Laghate, G. Hattab, D. Cabric, and S. Pollin, "Towards instantaneous collision and interference detection using in-band full duplex," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2017.

[73] Xilinx, "Spartan-3 FPGA family," visited on 27/04/2017. [Online]. Available: https://www.xilinx.com/products/silicon-devices/fpga/spartan-3.html

[74] N. H. Kuiper, "Tests concerning random points on a circle," *Indagationes Mathematicae (Proceedings)*, vol. 63, pp. 38 – 47, 1960.

[75] T. Vermeulen and S. Pollin, "Sense and abort: a real-time in-band full duplex collision detection network," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, Submitted.

[76] M. Schmidt, D. Block, and U. Meier, "Wireless interference identification with convolutional neural networks," *arXiv preprint arXiv:1703.00737*, 2017.

[77] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. Sukhavasi, C. Patel, and S. Geirhofer, "Network densification: the dominant theme for wireless evolution into 5g." *IEEE Communications Magazine*, vol. 52, no. 2, pp. 82–89, 2014.

[78] "Deliverable d1.1 deliverable d1.1: Scenarios, requirements and kpis for 5g mobile and wireless system," visited on 13/04/2017. [Online]. Available: https://www.metis2020.com/wp-content/uploads/deliverables/METIS_D1.1_v1.pdf

[79] K. Nishide, H. Kubo, R. Shinkuma, and T. Takahashi, "Detecting hidden and exposed terminal problems in densely deployed wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 11, pp. 3841–3849, November 2012.

[80] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics," *Communications, IEEE Transactions on*, vol. 23, no. 12, pp. 1400–1416, 1975.

[81] E. Ziouva and T. Antonakopoulos, "CSMA/CA performance under high traffic conditions: throughput and delay analysis," *Computer communications*, vol. 25, no. 3, pp. 313–321, 2002.

[82] Y. Liao, K. Bian, L. Song, and Z. Han, "Full-duplex MAC protocol design and analysis," *Communications Letters, IEEE*, vol. 19, no. 7, pp. 1185–1188, July 2015.

[83] T. Vermeulen, F. Rosas, B. van Liempd, M. Verhelst, and S. Pollin, "An energy-scalable in-band full duplex architecture," in *IEEE Internation Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, 2015.

[84] K. Jamieson and H. Balakrishnan, "PPR: Partial packet recovery for wireless networks," in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp. 409–420.

[85] F. Rosas, R. Souza, M. Pellenz, C. Oberli, G. Brante, M. Verhelst, and S. Pollin, "Optimizing the code rate of energy-constrained wireless communications with HARQ," *Wireless Communications, IEEE Transactions on*, vol. 15, no. 1, pp. 191–205, Jan 2016.

[86] F. Rosas and C. Oberli, "Modulation and snr optimization for achieving energy-efficient communications over short-range fading channels," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 12, pp. 4286–4295, 2012.

[87] National Instruments, "NI USRP-2942R - national instruments," visited on 13/01/2017. [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/nl/nid/212434

[88] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[89] M. Duarte, C. Dick, and A. Sabharwal, "Experiment-driven characterization of full-duplex wireless systems," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 12, pp. 4296–4307, December 2012.

[90] K.-S. Kim and K. Lee, "Low-power and area-efficient FIR filter implementation suitable for multiple taps," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 150–153, 2003.

[91] Atmel, "Atmega128rfa1 datasheet," visited on 13/04/2017. [Online]. Available: http://www.atmel.com/Images/Atmel-8266-MCU_Wireless-ATmega128RFA1_Datasheet.pdf

[92] "Gartner says 6.4 billion connected "things" will be in use in 2016, up 30 percent from 2015," visited on 13/01/2017. [Online]. Available: http://www.gartner.com/newsroom/id/3165317

[93] "Visual networking index (VNI)," visited on 13/01/2017. [Online]. Available: http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html

[94] P. Park, P. D. Marco, P. Soldati, C. Fischione, and K. H. Johansson, "A generalized markov chain model for effective analysis of slotted IEEE 802.15.4," in *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, Oct 2009, pp. 130–139.

[95] T. W. Anderson and D. A. Darling, "Asymptotic theory of certain" goodness of fit" criteria based on stochastic processes," *The annals of mathematical statistics*, pp. 193–212, 1952.

[96] B. van Liempd, B. Hershberg, K. Raczkowski, S. Ariumi, U. Karthaus, K.-F. Bink, and J. Craninckx, "A +70dbm IIP3 single-ended electrical-balance duplexer in 0.18 um SOI CMOS," in *Solid-State Circuits Conference-(ISSCC), 2015 IEEE International*. IEEE, 2015, pp. 32–33.

[97] B. C. Levy, *Principles of Signal Detection and Parameter Estimation*. Springer Publishing Company, Incorporated, 2008.

[98] L. Song, Y. Liao, K. Bian, L. Song, and Z. Han, "Cross-layer protocol design for CSMA/CD in full-duplex WiFi networks," *IEEE Communications Letters*, vol. PP, no. 99, pp. 1–1, 2016.

[99] R. Tandra and A. Sahai, "SNR walls for signal detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 4–17, Feb 2008.

[100] J. Zhang, F. Wang, Z. Zhong, and D. Cabric, "Local and cooperative spectrum sensing via kuiper's test," in *Communication Workshop (ICCW), 2015 IEEE International Conference on*, June 2015, pp. 579–584.

[101] A. N. Pettitt, "A two-sample anderson–darling rank statistic," *Biometrika*, vol. 63, no. 1, pp. 161–168, 1976. [Online]. Available: http://www.jstor.org/stable/2335097

[102] L. J. Gleser, "Exact power of goodness-of-fit tests of kolmogorov type for discontinuous distributions," *Journal of the American Statistical Association*, vol. 80, no. 392, pp. 954–958, 1985. [Online]. Available: http://www.jstor.org/stable/2288560

[103] S. Pollin, M. Timmers, and L. Van Der Perre, *Software defined radios: from smart(er) to cognitive*. Dordrecht: Springer, 2011.

[104] N. Flaherty, "Dresden sets up 5g communications research lab," visited on 13/01/2017. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1266564

[105] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving single channel, full duplex wireless communication," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 1–12.

[106] T. Schmid, O. Sekkat, and M. B. Srivastava, "An experimental study of network performance impact of increased latency in software defined radios," in *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*. ACM, 2007, pp. 59–66.

[107] C. Hunter, J. Camp, P. Murphy, A. Sabharwal, and C. Dick, "A flexible framework for wireless medium access protocols," in *Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on*. IEEE, 2006, pp. 2046–2050.

[108] D. Raychaudhuri, M. Ott, and I. Secker, "Orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*. IEEE, 2005, pp. 308–309.

[109] CREW project, "w-ilab.t (iminds) | CREW project," visited on 13/01/2017. [Online]. Available: http://www.crew-project.eu/wilabt

[110] J. Ansari, X. Zhang, A. Achtzehn, M. Petrova, and P. Mahonen, "A flexible MAC development framework for cognitive radio systems," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*. IEEE, 2011, pp. 156–161.

[111] D. van den Akker and C. Blondia, "MultiMAC: A multiple MAC network stack architecture for TinyOS," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*. IEEE, 2012, pp. 1–5.

[112] FLAVIA, "FLAVIA home page," visited on 13/01/2017. [Online]. Available: http://www.ict-flavia.eu/

[113] Zigduino, "Zigduino r1 manual," visited on 13/01/2017. [Online]. Available: http://wiki.logos-electro.com/zigduino-manual

[114] S. Yun, D. Kim, and L. Qiu, "Fine-grained spectrum adaptation in wifi networks," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 327–338.

[115] H.-S. W. So, G. Nguyen, and J. Walrand, "Practical synchronization techniques for multi-channel MAC," in *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, 2006, pp. 134–145.

[116] L. Tytgat, O. Yaron, S. Pollin, I. Moerman, and P. Demeester, "Avoiding collisions between IEEE 802.11 and IEEE 802.15. 4 through coexistence aware clear channel assessment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–15, 2012.

[117] CREW project, "Project overview | CREW project," visited on 13/01/2017. [Online]. Available: http://www.crew-project.eu

[118] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Signals, systems and computers, 2004. Conference record of the thirty-eighth Asilomar conference on*, vol. 1. Ieee, 2004, pp. 772–776.

[119] J. Walrand, *Communication Networks: A Concise Introduction*. Morgan and Claypool Publishers, 2010.

[120] B. Latré, P. De Mil, I. Moerman, N. Van Dierdonck, B. Dhoedt, and P. Demeester, "Throughput and delay analysis of unslotted IEEE 802.15.4," *Journal of Networks*, vol. 1, no. 1, pp. 20–28, 2006. [Online]. Available: http://dx.doi.org/1854/7653

[121] W. Segata and R. Lo Cigno, "On the feasibility of collision detection in full-duplex 802.11 radio," in *2017 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, Feb 2017, pp. 9–16.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
NETWORKED SYSTEMS, TELEMIC
Kasteelpark Arenberg 10 box 2444
B-3001 Leuven
tom.vermeulen@kuleuven.be